A LIBRARY OF SOFTWARE FOR THE **IBM PC** & **PCjr**

PC DISK

SIX NEW PROGRAMS ON DISKETTE

USER'S MANUAL VOL. 1 NO. 10 $34.95

FEATURING:

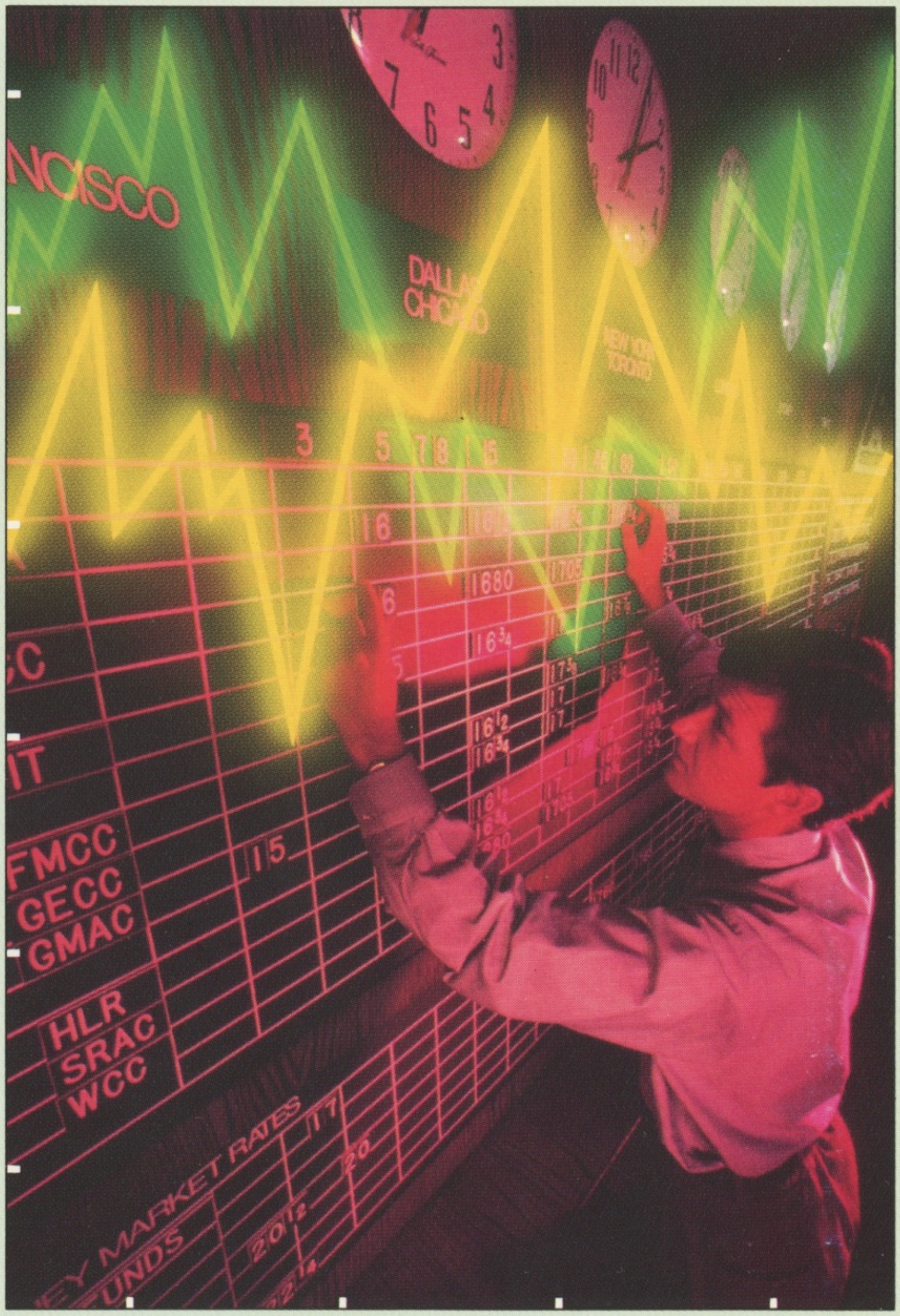
STOCK/BOND ANALYZER

IMPROVE YOUR BUY/SELL DECISIONS THROUGH PERFORMANCE TRACKING

PLUS:

- STOCK/BOND DATA FILE
- STATISTICS LIBRARY 2
- GRAPHICS TUTOR
- PICTURE PAINTER
- TIC TAC GO

DETAILS ON BACK COVER



A LIBRARY OF SOFTWARE FOR THE **IBM PC** & **PCjr**

PC DISK

PUBLISHER
André M. J. Van Hattum

EDITORIAL
Editor: Morris L. Effron
Software Editor: Dale Benzer
Managing Editor: Bohdan "Bo" Shmorhay
Documentation Editor: Bruce L. Murphy
Copyeditor: Laura Ninger
Assistant Editors: Carmine Coletti, Anne Condon, Matt Dolph, Evelyn Leon, Quadratec, Inc.
Production Coordinator: Eve Hinderer
Editorial Secretary: Lillian Pearsall

ART & DESIGN
Cristina Botta, Borys Patchowsky
Susan Molloy, Thomas Stvan.
Cover Photography: Michael Salas/Image Bank *Special Effects:* Photo Craft

COMPUTER PUBLICATIONS DIVISION
General Manager: Kenneth H. Koppel
Senior Vice President: Eileen G. Markowitz
Vice President, Editorial Director: Jonathan Lazarus
Vice President, Licensing and Special Projects: Jerry Schneider
Vice President, Creative Services: Herbert Stern
Circulation Director: Alicia Marie Ivans
Creative Director: Peter J. Blank
Editorial Director: Ernest F. Baxter
Business Manager: Gary A. Gustafson
Marketing Manager: Ronni Sonnenberg

ZIFF-DAVIS PUBLISHING COMPANY
President: Richard P. Friese *President, Consumer Magazine Division:* Albert S. Traina *Executive Vice President, Marketing and Circulation:* Paul Chook *Senior Vice Presidents:* Philip T. Heffernan, Sidney Holtz, Edward D. Muhlfeld, Philip Sine *Vice Presidents:* Baird Davis, George Morrissey, Rory Parisi, William L. Phillips *Treasurer:* Selwyn Taubman *Secretary:* Bertram A. Abrams
PC Disk is published nine times per year, by Ziff-Davis Publishing Company, One Park Avenue, New York, NY 10016. Subscriptions are available at $149.00 for six issues. POSTMASTER: Send address changes to *PC Disk*, P.O. Box 5930, Cherry Hill, NJ 08034
Editorial and Business Office: One Park Avenue, New York, NY 10016. Editorial 212-503-3300. For subscription inquiries and service write to *PC Disk*, P.O. Box 5930, Cherry Hill, NJ 08034

PC Disk is an independent journal, not affiliated in any way with International Business Machines Corporation. IBM is a registered trademark of International Business Machines Corp. Entire contents copyright © 1984 Ziff-Davis Publishing Company. All rights reserved; reproduction in whole or in part without permission is prohibited.

ISBN #087165-110-6 PC DISK VOLUME 1 NUMBER 10

CONTENTS

BUSINESS

EDUCATION

UTILITIES

RECREATION

BACK ISSUES OF .... PC DISK

Add to your *PC Disk* collection today, while copies are still available. The more complete your library of back issues, the more enjoyment and use you can get from your personal computer.

The currently available back issues begin with Volume 1, Number 1. Order any issue you want, being sure to specify volume and issue number and the quantity for each. This offer is subject to availability. If a particular issue is out of stock, your payment will be refunded promptly.

Back issues of *PC Disk* are priced at $31.95 each, which includes postage and handling. Please write to:

PC Disk
P.O. Box 5930
Cherry Hill, NJ 08034

Be sure to include volume and issue number, as well as the number of copies desired, and the address to which you would like those copies sent. Please enclose your payment with your order. For faster service, call:
800-932-0017
in NJ 800-624-0497

DEPARTMENTS

TERMS AND CONDITIONS

LIMITED WARRANTY

In the unlikely event that the disk contained in this issue is defective in manufacture, return it within 30 days after purchase or receipt to *PC Disk*, P.O. Box 5930, Cherry Hill, NJ 08034, and it will be replaced at no charge. Merchandise for exchange must be accompanied by proof of purchase or a subscription label.

The publisher shall have no liability or responsibility to any purchaser or subscriber or to any other person or entity with respect to any liability, loss, or damage directly or indirectly arising out of the use of the disk and the programs contained in this issue, including, but not limited to, any loss of business or any other incidental or consequential damages.

COPYRIGHT

Programs and Documentation Copyright © 1984 Ziff-Davis Publishing Company. All rights of reproduction in all forms and media strictly reserved.

The original purchaser/subscriber only is authorized to make copies solely for use in his/her own computer. Distribution, transmission, or transfer of copies to any other person is an infringement of copyright.

FROM THE EDITOR

Hello, do I know you? The obvious answer is no—we've never met, and have probably never spoken on the phone or corresponded. But I do know a lot about you; not *you* personally, but you, our readers. I know that you're probably over 25 years of age, well-educated, and engaged in a well-paying, upwardly mobile career. I also know that chances are good that you have (or will have) a family. Most importantly, I know that you own a microcomputer, and that you're interested in what a microcomputer can do for you.

That's research for you, and it is helpful. It gives us an idea of how to market our product, how to give it an appealing look, and how to sell it. It also gives us some ideas about what kinds of programs we should publish. But even the best research, and Ziff-Davis does have considerable research expertise, has two shortcomings: it deals in averages and generalities, of statistical necessity; and while it's good at telling you what's going on, it has difficulty determining why.

These problems of specificity and motivation are particularly troublesome to us in editorial. In order to give you a product of value and utility, we need to know what you, both individually and collectively, want. We have certain general guidelines which represent our basic editorial concept. We want to publish programs that are both practical and easy to use, and we want to include a variety of such programs across several application categories each issue, so that *PC Disk* helps you get more from your micro. A noble philosophy, but what about the specifics?

We know you own a microcomputer, but why did you buy it? Did you buy it to learn more about computers, or because you considered it a useful tool? Did you buy it primarily for your own use, or for use by your spouse and children as well? Did you buy it to create your own custom software, or to treat it as a versatile appliance by using existing software?

We know you're interested in using a microcomputer, but how do you want to use it? Is it just for business applications, or are personal and home uses a priority? Are you interested in learning the nitty-gritty of how the machine works, or in simply getting it to do the things you want done? How much time do you want to devote to using your computer?

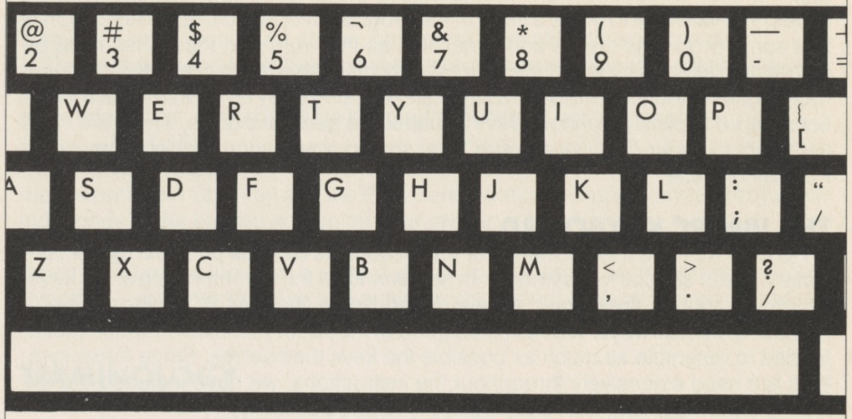
The answers to these motivational questions are very important to us. Getting specific feedback is also important: which of our programs did you like, and which didn't you like? Which were easy to understand and use, and which weren't? What programs would you like to see us publish? And in all cases, why?

These questions are not a plaintive appeal for guidance. We certainly have some well-considered ideas concerning the answers. On the other hand, these questions are not rhetorical either. Research takes us only so far. We still have a lot to gain by hearing from you personally. While we cannot be all things to all people, and don't intend to try, your input can help us make *PC Disk* a more satisfying product. So don't be reticent—let us have the benefit of your thoughts on these key editorial questions.

Hello, it's great to meet you.

Morris L. Effron

TECHNICAL PREFACE



To help our readers make the most of PC Disk, *we would like to provide some background information concerning the editorial diskette, the accompanying manual, and how to use both. We don't expect all of the following topics to interest all our readers. Nevertheless, we prefer to err on the side of comprehensive support rather than leave any of our readers confused. So we encourage everyone to at least skim this section to assure a solid background for the use of* PC Disk.

USAGE REQUIREMENTS

PC Disk has been designed for use on an IBM Personal Computer with a minimal set of hardware components: a keyboard, a monitor, and the computer itself. Running DOS 1.1, a minimum of 64K of main memory is required. Under DOS 2.0 and DOS 2.1, a minimum of 128K is required. The display unit can be a monochrome display adapter and monitor, or the color graphics display adapter with a color monitor, a black and white monitor, or an RF Modulator and TV set. The computer itself can be the PC, the PC-XT, or the PCjr. For the PCjr we recommend the use of a monitor rather than a TV set, since most of our software is written for an 80-column display. However, a TV set does provide a workable display.

These three pieces of equipment are all you need to run the majority of *PC Disk* software. Wherever possible, we make the use of any other hardware optional. So, for example, many of the programs will generate printed output, but a printer is not required in order to use them. Occasionally, however, due to the nature of a program or its design, a particular piece of equipment will be necessary. When a program requires equipment not in the minimum configuration stated above, this component will be listed as a "Special Requirement" on the program's title page in this manual.

In regard to software, all *PC Disk* programs are designed to run under DOS 1.1, DOS 2.0, and DOS 2.1. Furthermore, all BASIC programs in the magazine are designed to run under Microsoft's Advanced BASIC. Neither DOS nor Advanced

BASIC are provided on the *PC Disk* diskette; they must be acquired separately. As a rule, these are the only outside software elements you will need to use *PC Disk*. We will occasionally publish a program which uses some additional, publicly available software product. Any such additional software will be listed as a "Special Requirement" on the program's title page in this manual.

Our closing remark on this topic is a recommendation that you make a copy of your *PC Disk* diskette to work with and save the original as a backup. In some cases, you will have to make a copy of the program in order to use it. The reason is that some programs create additional files as they run, and these files must be stored on diskette as well. Since your *PC Disk* diskette is write-protected, it cannot receive these additional files, so a separate, working copy is needed. These situations will be explicitly mentioned in the manual. In general though, where the manual refers to "your *PC Disk* diskette" you should read "your working copy of the *PC Disk* diskette."

THE IBM PC KEYBOARD

In *PC Disk* we have tried to make our instructions as clear as possible by the consistent use of special key symbols. In addition to all the common typewriter keys, which we print as they would appear when typed, the IBM PC keyboard has a number of special keys. We have designed symbols for these keys, which are intended to resemble as much as possible the keys themselves. Since these symbols are used extensively throughout the instructions, we have included the following list to help you, our reader, get any needed bearings.

THE FUNCTION KEYS	F6
THE ESCAPE KEY	Esc
THE CONTROL KEY	Ctrl
THE TAB KEY	|← →|
THE BACKSPACE KEY	←
THE SHIFT KEY	⬆
THE PRINT SCREEN KEY	PrtSc *
THE ENTER KEY	↵
THE NUM LOCK KEY	Num Lock
THE CURSOR CONTROL KEYS	2 ↓ 4 ← 6 → 8 ↑
THE INSERT AND DELETE KEYS	0 Ins . Del
THE CAPS LOCK KEY	Caps Lock
THE CONTROL AND SCROLL LOCK KEYS	Ctrl Scroll Lock

ATTENTION PCjr OWNERS:

The instructions in our manual are written for the PC keyboard. Section 4 of your "Guide to Operations" manual for the IBM PCjr provides complete information on how to translate PC keystrokes to PCjr keystrokes. Please refer to this section for guidance in following our operating instructions.

TEXT CONVENTIONS

Most of the textual conventions are fairly obvious. Besides the special key symbols and names discussed above, command lines also deserve mention.

The lines set apart from the narrative text are commands that should be typed in exactly as they appear. When two key symbols appear next to each other in such a command line, they should be pressed simultaneously. For example:

⬆ PrtSc *

means press the Shift key and the Print Screen key simultaneously, thereby printing a copy of the current screen on your printer.

There is one exception to typing in command lines exactly as they appear. When a command includes a word such as "somename" or "programname" or "yourfile", you should, when you enter the command, replace that word (but not any punctuation) with a valid filename of your choice.

TERMINOLOGY

In the preceding section we identified the special key symbols used in this manual, and gave a name to each one. For example:

↵

is called the Enter key. In our instructional narrative, it sometimes makes more sense to refer to a special key by its name than by its symbol. Thus the key names in the preceding section are also special terms for the purposes of this manual. Familiarize yourself with the names to facilitate your use of the manual, and refer to the preceding section as a glossary of key names when necessary.

In addition to the key names, a few other terms and phrases which are used in this manual may be unfamiliar to you.

We commonly speak of putting a diskette in the "default drive". This may seem like a needlessly vague phrase. After all, we know a diskette drive always has a one-letter identifier associated with it, so why not refer explicitly to that letter? The problem with using an explicit letter reference is that it can create confusion about what exactly you must do. In other words, operationally it does not matter whether you put the diskette in the A Drive, the B Drive, or even the C Drive (if you have a third diskette drive). What matters is that you put the diskette *in the drive that is currently active*, i.e. the drive whose letter prompt currently appears on the screen. This is your "default drive" because any disk command without a drive letter will look at the diskette in this active drive. So when you put a diskette in the "default drive", you can then issue commands referencing that diskette without the use of letter identifiers.

Every start-up procedure for a BASIC program requires you to "Load Advanced BASIC into your PC". To run a *PC Disk* BASIC program, the BASIC Interpreter must be up and running on your machine—you must be "in BASIC". BASIC is really a program like any other. To start it, you must load it from a disk into your PC and start it running. This is precisely what happens when you put your DOS disk (or any disk with the file BASICA.COM) in the default drive and type:

```
BASICA ↵
```

By so doing you "Load Advanced BASIC into your PC".

PROBLEM HANDLING

There really is not a great deal to say about problem handling with *PC Disk*. If you use this software on the right equipment running the appropriate system software, as outlined in the Technical Preface, you should experience no problems. Nevertheless, a few comments may resolve some more obvious difficulties.

Any BASIC program can be interrupted at any time by pressing:

Ctrl Scroll Lock

If you do not see the "Ok" message immediately, indicating that you are back in BASIC, press these keys again. This is a rather drastic but effective way of regaining control of the computer. You won't damage any of the programs in this way, since they're still intact on the diskette. However, you may lose data you entered while the program was running.

If you interrupt a program, you may find that the function keys no longer perform as they had before you started the program. This is because many *PC Disk* programs reset the function keys during execution, then restore the original settings upon completion. An interrupt causes an abnormal termination of a program, so the function keys are not restored. To correct this situation, simply exit from BASIC and then return to BASIC.

You may find at times that the cursor control keys are not working as they should. This is because the keys are not in cursor control mode. The key that switches these keys between numeric mode and cursor control mode is the Num Lock key. So to restore the keys to cursor control mode, press:

Num Lock

If you try to send something to the printer when there is no printer, or when the printer is off or offline, you can hang the system. If the computer does not put out any message, but just remains idle, you will have to say good-bye to whatever you were doing and re-boot your system.

Should you ever find a "bug" in a *PC Disk* program, *please do not send back your diskette*. Call:

212-503-5330

or write:

PC Disk
Problem Handling
One Park Avenue
New York, NY 10016

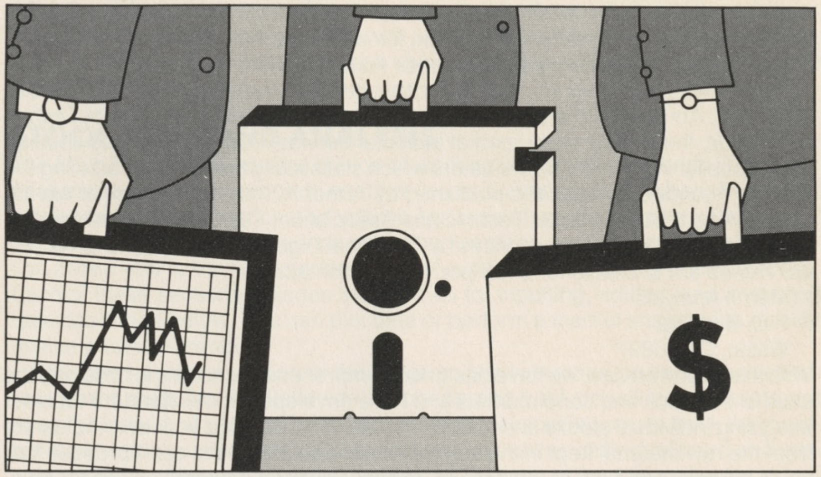
Remember to include your name, address, and telephone number, as well as a thorough description of your system and the problem you have encountered.

If your disk is *physically* damaged, return it, with a note, to:

PC Disk
P.O. Box 5930
Cherry Hill, NJ 08034

Feel free to contact us at any time. In most cases, we can offer a solution to your problem that will save you time and inconvenience.

BUSINESS



STOCK/BOND ANALYZER

By Doug Henwood

Special Requirements: None
Files Used: STOKBOND.BAS
STOKBOND.OVR
STOCK.DAT
BOND.DAT
Files Created: STOCK.HIS

Once upon a time, technical analysis of the stock and bond markets was the province of a small coterie of investors who kept their charts and graphs by hand. Their colleagues looked upon them with suspicion—could the insights they gained possibly be worth all those hours tracing out mysterious lines on graph paper?

Then the wealthy investment groups discovered that they could computerize all that pencil and paper work. At first, large mainframes made technical analysis more respectable, but using them was too expensive for individual investors. Now, however, the microcomputer allows the individual investor to use the same tools as the investing conglomerates. One of these valuable investment tools is STOCK/BOND ANALYZER, *the first part of a complete security analysis and accounting system. Whether you're a long-term investor or a nimble trader, a novice or a sophisticated investor,* STOCK/BOND ANALYZER *will be a powerful aid in helping you understand the markets.*

BACKGROUND

Most investors distinguish between technical and fundamental analysis of the financial markets. Fundamental analysis focuses on broad economic trends, such as the condition of the economy and the role of the business cycle, and specific details about a company, such as the quality of its management and the standing of its products or services in the marketplace. While technicians do not dismiss these fundamentals, they believe that one can learn more about markets by watching the markets themselves, because by the time fundamental facts become known about a market or a stock, it may be too late. *STOCK/BOND ANALYZER* is intended to help you read these market signals.

In compiling this program, we have consulted with many experts in the investment field. Among them are John R. McGinley, Jr., an Associate Editor of *Technical Trends,* the weekly stock market indicator service. *Technical Trends* specializes in publishing only those indicators which statistically test well in predicting the market. Founded in 1960 and published by Arthur A. Merrill, recipient of the 4th annual Award of the Market Technicians Association, *Technical Trends* was selected as one of the top ten services by Technical Digest. Subscriptions to *Technical Trends* are available for $125 per year by contacting:

Merrill Analysis
P.O. Box 792
Wilton, CT 06897

With this expert advice, we have designed comprehensive models for tracking the state of the stock and bond markets and have developed some standard tools for analyzing individual stocks and bonds. You can also develop your own indicators from the raw material kept in the market model data base. We can't promise that you'll become rich from its indications, but you will get a powerful device for gaining insight into today's volatile markets.

This is the first in a series of investment analysis and record-keeping programs. The next program in the series will provide analysis of futures and options, while the final one will be a complete portfolio accounting system. Each module will stand alone, but since they can share data with each other, they can be integrated into a powerful and informative package.

PROGRAM STRUCTURE

The *STOCK/BOND ANALYZER* program has a parallel tree structure. There are two main branches, or analysis routines: one for stocks and one for bonds. The flow of command for each of these routines is similar, except where market-specific differences arise. This means that the sequences for entering, updating, and graphing data operate the same for stocks as for bonds. The routines are also parallel in that they both allow you to see a market model, take a market snapshot, and analyze an individual stock or bond.

START-UP

Before running *STOCK/BOND ANALYZER*, transfer the necessary files from your *PC Disk* diskette onto a blank, formatted diskette. To make the copy, make sure your system is in DOS. Then put your *PC Disk* diskette in the default drive and type:

```
COPY  STOKBOND.*  B: ↵
```

where "B:" is the drive containing your blank, formatted diskette (on a single-drive system, DOS will prompt you to swap in your destination work diskette for each COPY command). Repeat this procedure for the data files contained in the *STOCK BOND DATA FILE* program, STOCK.DAT and BOND.DAT, if you wish to continue recording data along with these historical records.

To start *STOCK/BOND ANALYZER*, insert a system disk in your default drive and load Advanced BASIC into your PC by typing:

```
BASICA/S:160 ↵
```

Then place your work copy of the program into your default drive and type:

```
RUN “STOKBOND ↵
```

After the Title Screen is displayed, you must specify which disk drive contains your data files. Next, the Main Menu will appear. From this menu, you can choose to work with stocks or bonds. A third choice allows you to exit the program and return to DOS.

STOCK AND BOND ANALYSIS

The Main Menu gives you the choice of working with stocks or bonds. The stock analysis section of the program enables you to examine a comprehensive market model and obtain the price and volume histories of up to 20 stocks. The market model gives you an overall sense of the market place, while the stock history section allows you to track the performance of individual stocks or groups of stocks. Each of these sections includes the routines for inputting, editing, and analyzing data described below. You can plot data or perform a market snapshot to quickly evaluate current trends.

The bond analysis section of the program has two major parts: one which allows you to analyze a market model for bonds, and a set of routines which lets you examine individual bonds. The bond market model enables you to track the major bond indicators, while the additional routines analyze an individual bond and provide you with yield and accrued interest calculations. An additional utility calculates the number of days between two dates.

ENTERING, DISPLAYING, AND EDITING DATA

The routines for entering, displaying, and editing data are identical for both the stock and bond modules. When you first start the program, you can add data onto historical stock and bond files from the *STOCK/BOND DATA FILE* program contained in this issue, or begin a new data base. First choose the Enter/ Display/Edit option and you will be prompted to enter a date. Type the month, day, and year separated by any single characters (spaces, slashes, or hyphens are all acceptable). Use two digits for years between 1900 and 1999, and all four digits for years after 2000. When you are adding stock data, the date will correspond to a single day's information, while a date entered for bond data will represent an end-of-week value for bond indicators.

If you entered a date that was already in your data base, the program assumes you want to display or edit the data. If you choose a date after the last date in the file, the program will assume you are entering new data.

When you are displaying old data, press the F1 key to change any field, the F2 key to be prompted for a new date, or the Escape key to return to the previous menu. When you are entering data, after you finish with the last field, you can press the F1 key to change the data, the F2 key to write the new data to the disk and be prompted for a new date, or the Escape key to return to the previous menu without writing the new data to the disk. To edit data, use the arrow keys to move to the field you want to change, and enter the new value. The old value will disappear when you begin to type. When you are finished editing, press the F2 key to write the new data to the disk, or the Escape key to return to the previous menu without writing the new data to the disk.

Throughout the program, you may express fractional values as decimals, fractions, or 32nds by using the decimal point, slash, or apostrophe, respectively. For example, 32.25, 32 1/4, and 32 '8 represent the same value. The program uses decimal notation only when it displays values, but you will find it easier not to have to convert fractions when entering information.

GRAPHING DATA

There are three types of graphs used in the program, although all types are not available in every section. The first type is the traditional price and volume graph with a moving average. (This type is used in the stock history and stock model sections.) The second is a single indicator with a moving average graphed on a full screen; the third is two indicators graphed together, each occupying half the screen. The price/volume graph is appropriate for graphing stocks and market averages; the single indicator type is useful for graphing oscillators like TRIN and net free reserves, where relatively fine distinctions may be of great importance; the third graph type is best for graphing one market indicator and an index, to check for correlations between market movement and changes in the indicator.

If you choose either of the last two graph types, you must specify whether you want to graph the indicator itself or its rate of change. Rate of change (also called momentum) measures the percentage difference in an indicator from a fixed prior date. For example, the ten-day rate of change on the Dow Jones average measures the difference between the average on a given day and its price ten days earlier. When graphing a moving average or rate of change, you will be asked for a term; enter the number of days you want. Ten-day values are commonly used for short-term analysis; 200 days, for longer-term analysis.

After you choose the graph type, you will be presented with a list of indicators to be graphed. With the models, this list will include all the indicators kept in the data base, with "Define your own indicator" as the last option. If you choose this option, you will be asked for a description, which you enter using conventional algebraic notation. Examples are given in each section below.

Once the graph is on the screen, you have the option of writing the graph data into a separate file. This output file can be a regular text file which you can later print or edit, or a DIF file, which can then be used by a spreadsheet or other DIF-compatible program.

STOCK MARKET MODEL

There are two options within the stock section of the program. These are the stock market model and the individual stock price histories. To get a general feel for the market, choose the market model. Your options within this section are to input new data, display or edit data, graph or analyze old data, and perform a market snapshot.

The stock market model is a set of widely followed market indicators which, when properly analyzed, can give you a good short- to intermediate-term feel for the market. This model is intended for stock, option, and futures traders who want to catch the twists and turns of the market—traders whose time frame is measured in days and weeks, not months and years. All indicators we have included are available in the Friday edition of the *Wall Street Journal*, in addition to other public sources. The model is not designed to interpret data; it is meant to supply the market indicators for you to analyze. If you are seriously interested in this model, you should enter the data daily. Otherwise, the calculations become meaningless. The values used in the stock market model are:

1) Dow Jones Industrial average
2) Standard & Poors (S & P) 500 Stock index
3) The nearest S & P 500 index future
4) NYSE (New York Stock Exchange) composite index

5) The nearest NYSE index future
6) Value Line index
7) The nearest Value Line index future
8) NYSE total volume (not including regional exchanges)
9) NYSE advancing volume
10) NYSE declining volume
11) NYSE stocks advancing
12) NYSE stocks declining
13) NYSE stocks unchanged
14) CBOE (Chicago Board Options Exchange) call volume
15) CBOE put volume

Data for numbers 8-10 should be entered without the last three zeros (e.g. 100253 for 100,253,000). Descriptions of each of these indicators appear in the *STOCK/BOND DATA FILE* article contained in this manual.

You can find the data for these indicators in a variety of newspapers. The Dow Jones Industrial Average is prominently displayed in all of them; the values for numbers 2-7 can be found on the futures pages of your newspaper. Use the closing price for the nearest future, except during delivery month, which is the month in which a contract expires. For example, in July, you would use the September contract quote, but on the first of September, you would begin using the December contract. Indicators 8-12 appear on the inside back page of the *Wall Street Journal* and on the first page of the NYSE listings in the *New York Times* (the page with the chart of the NYSE composite). Finally, you can locate the values for numbers 14-15 on the options pages of both newspapers, at the end of the complete CBOE listing.

Most of the market indicators are stored in your data base as you input them, although the program computes some values from your raw data. From the advance/decline data, the program computes and stores the short-term trading index, known as TRIN or the Arms index. The exact formula is:

[advances/declines]/[advancing volume/declining volume]

Readings below 1 are bullish, and those above 1 are bearish. Raw data for advances and declines is translated into a cumulative advance-decline figure by adding the net difference between advances and declines to a cumulative figure. What is important is the direction in which this indicator is moving, not its absolute level.

MARKET SNAPSHOT

You can easily obtain a quick picture of some crucial indicators within the market model by taking a market snapshot. When you are prompted, either specify a date or just press the Enter key to see the latest data. The report includes values for all the averages as well as their five- and ten-day rates of change; volume, ten-day average volume, and ten-day cumulative up volume; the total of all the net differences between advances and declines for the previous ten days; and the ten-day average of TRIN. You can use this feature not only to get a picture of a current market, but also to inspect some crucial past turns in the market and compare them to the current situation.

GRAPHING DATA: THE STOCK MARKET MODEL

The stock market model uses all three of the graphs discussed above: the price/volume graph, the single indicator on a full screen, and two indicators on a single screen. After you select the graph type, you must choose the values you want to graph from a list of appropriate indicators.

If you use one of the last two graph types, you will notice that "define your own indicator'' is the last choice. This means that, by using algebraic notation, you can describe a value to graph by combining the numbers representing the other indicators. For example, entering V1/V2 will create a graph of the ratio between the Dow Jones average and the S&P 500; (V7-V6)/V6 will create a graph of the premium of the Value Line future over its actual or ''cash'' value. You can be as complex as you wish; just remember that algebraic rules of precedence apply.

STOCK PRICE HISTORY

The second option in the stock market section of *STOCK/BOND ANALYZER* lets you record daily price and volume histories for up to 20 different stocks. While this section of the program is designed for individual stock histories, you can also track futures contracts and indicators not covered in the market model (e.g., the Amex and NASDAQ indexes with respective volumes, or financially-oriented futures contracts like bonds and gold.) Note that when you are tracking futures contracts, you should use open interest figures in place of volume.

With this module of the program, you can set up a new stock history, enter new data, edit or display old data, edit or display head information, graph or display old data, analyze an individual stock, or adjust a stock price for a split.

SETTING UP A STOCK HISTORY

In order to analyze stocks with this program, you must first enter some introductory information about each one. This information includes the stock symbol, a name (fewer than 16 characters), the current dividend and latest 12-months earnings (use a minus sign for a loss), some indicators used in modern portfolio analysis (alpha, beta, and volatility), and the year's high and low prices. (Use a decimal for volatility, e.g. .21 for 21%.) You can get the dividend figure and the initial values for high and low from the newspaper; earnings, alpha, and beta are available from brokers, data bases, and information services. These numbers will be used mostly by the other programs in the investment management series (*OPTIONS/ FUTURES ANALYZER* and *PORTFOLIO ANALYZER*) for computing portfolio risk/return and theoretical call prices. If you don't have all of the numbers, press the Enter key without entering any values; you can always add them later.

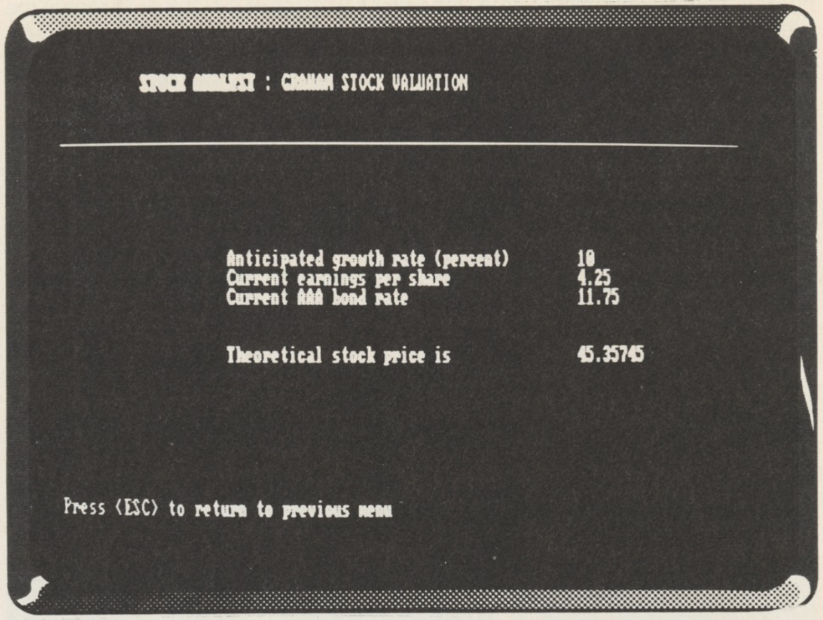
Use the standard ticker symbol for a stock to ensure compatibility with the other programs in the series. You can indicate preferreds with a slash—e.g., T/A would be AT&T preferred class A.

EDITING OR DISPLAYING HEAD INFORMATION

Use this feature to get a quick statistical picture of an individual stock or to update the values on file. The program will present a list of the stocks in your data base, and will ask you to choose one. Press the Enter key to return to the previous menu. After you select a stock, the information described in ''Setting Up a Stock History'' is displayed. You can change any of the information shown by pressing F1, as described above.

ANALYZING AN INDIVIDUAL STOCK

This feature provides you with a theoretical valuation of a stock according to the Graham valuation model, which is used by professional portfolio managers. You will be prompted for the percentage growth rate, earnings per share, and the current AAA corporate bond rate. The program will respond with a theoretical stock price. In general, a stock priced below its theoretical rate is underpriced and one priced over its theoretical price is too rich—but there can never be any assurance that the markets will heed this theory and quickly reprice the stock.



```
STOCK ANALYST : GRAHAM STOCK VALUATION

                    Anticipated growth rate (percent)      10
                    Current earnings per share             4.25
                    Current AAA bond rate                  11.75

                    Theoretical stock price is             45.35745

Press <ESC> to return to previous menu
```

Analyze an individual stock

ADJUSTING PRICE HISTORY FOR SPLIT

When a stock splits, its historical price and volume data are readjusted to assure consistency over time. When you select this option, the list of stocks is displayed and you are prompted to enter the number of the stock being adjusted. Next enter the split fraction as it is usually described. For example, a 3-for-2 split is entered as 3/2; a 1-for-10 reverse split would be entered as 1/10. After you have entered the fraction, the program will return to the Stock Price History Menu. Be careful when adjusting the same stock more than once, because repeated rounding may make your price outcomes inexact.

GRAPHING DATA

This option allows you to create a price/volume graph of any stock in your 20-stock data base. If you choose this option, you will be presented with a list of the available stocks; choose the number corresponding to the stock you want to graph. (The other graph types are not available in this section.)

BOND MARKET MODEL

Like the stock market model, this model gives you a comprehensive picture of the state of the interest-rate markets. All of the bond market information is available from the daily newspaper, except M-1 money supply and net free reserves. The money supply is reported on Thursday afternoons, and appears in Friday's newspaper under a title like "Federal Reserve Report". The free reserves number is reported every two weeks, also on Thursday, and also in the Fed report. If there is no new reserve information, press the Enter key to leave it blank or to repeat the previous week's number. (Free reserves are also referred to as borrowed reserves, especially if they are negative. Use a minus sign to express a negative value, not parentheses.) For the municipal bond rate, use a consistent interest rate index, such as the Bond Buyer index, the Merrill Lynch index, or the Kenny index. It doesn't matter which rate you use; just be consistent. For the treasury rates, try to use yield figures for bonds trading close to 100, and avoid older, heavily discounted bonds.

This model should be updated weekly. Since the Fed data appears in Friday's newspapers, this is probably the best day to update your model.

The components of the market model are:

1) discount rate
2) federal funds rate
3) commercial paper (nonfinancial)
4) interest rates on 3-month T-bills
5) interest rates on 6-month T-bills
6) interest rates on 1-year T-bills
7) interest rates on 2-year T-notes
8) interest rates on 3-year T-notes
9) interest rates on 5-year T-notes
10) interest rates on 7-year T-notes
11) interest rates on 10-year T-notes
12) interest rates on 20-year T-bonds
13) interest rates on 30-year T-bonds
14) Municipal bond rate
15) net free reserves
16) M-1 money supply

Enter data for numbers 14 and 15 as billions, e.g. 525.3 for 525,300,000,000.

Like the stock indicators, these bond indicators are fully explained in the *STOCK/BOND DATA FILE* article in this manual. The bond market model contains options to input, display, and edit data; take a market snapshot; or graph and analyze data. The operating sequences for these four options are identical to those used for the stock market model.

MARKET SNAPSHOT

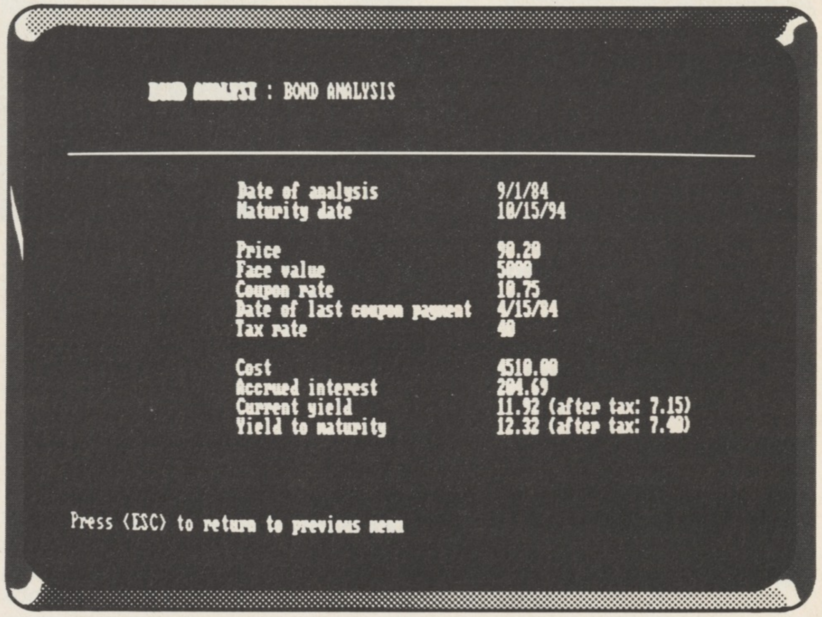
Just as you could for the stock market, you can use the snapshot feature for the bond market to look at some crucial indicators. You will be asked for a date; if you want to see the latest information, then press Enter without specifying a date. The program will then display the values for some crucial indicators, including the spread between the federal funds rate and the discount rate, the spread between commercial paper and 3-month T-bill rates, the spread between the 3-month T-bill and the 30-year bond, the spread between municipal rate and the 30-year T-bond, and the 4-week average of net free/net borrowed reserves. After this is shown, you will be asked if you want to see a yield curve, which is a graph of the yields available on Treasury securities for each maturity in the data base. Answer "Y" to the prompt in order to display the yield curve; answer "N" to return to the date prompt.

GRAPHING DATA: THE BOND MARKET

The two graph types available in the Bond Model are the single indicator and double indicator varieties. (Since there is no volume data, the price/volume graph is irrelevant.) After choosing the graph type, you will be asked to select the indicator(s) you want displayed. As in the stock model, this section includes the option of describing your own indicator. To see a graph of the spread between the Federal Funds rate and the discount rate, enter "2-1" when prompted for a description; "16-13" will describe the spread between the municipal bond rate and the 30-year Treasury bond rate. Again, you can get as complex as you like within the limitations of algebraic notation.

ANALYZING AN INDIVIDUAL BOND

The second option available in the bond analysis portion of the program lets you analyze an individual bond. This feature is independent of the other program functions because it does not access any of the data files previously established. You must enter all the pertinent information at the time of the analysis. The bond analysis routines will calculate the yield information given a price, or it will calculate price from yield information.



```
BOND ANALYST : BOND ANALYSIS

              Date of analysis              9/1/84
              Maturity date                 10/15/94

              Price                         90.20
              Face value                    5000
              Coupon rate                   10.75
              Date of last coupon payment   4/15/84
              Tax rate                      40

              Cost                          4510.00
              Accrued interest              204.69
              Current yield                 11.92 (after tax: 7.15)
              Yield to maturity             12.32 (after tax: 7.40)


Press (ESC) to return to previous menu
```

Analyze an individual bond

If you choose to perform a yield calculation, you must specify whether you are working with a treasury bill or a bond. In either case, you will be prompted for the price, face value, current date, maturity date, and your marginal tax rate for a security of this type (U.S., local or non-local municipal, or corporate). If the security is a bond, you will also be asked for the coupon rate and date of last coupon payment. (For zero coupons, enter a 0. Otherwise, the program assumes semiannual interest payments.) For treasury bills, the program will respond with the total cost, bond-equivalent yield and the after-tax yield. For bonds, you will be shown the cost, accrued interest, total cost (cost plus accrued interest), current before- and after-tax yields, and the before- and after-tax yield to maturity.

The second option in the bond analysis section enables you to calculate the price of a bond given prevailing interest rates. If you choose this option, you will be prompted for the date of analysis, the coupon rate of the bond, the date of maturity, and the prevailing interest rates on similar securities. The program will respond with the price of the bond on the chosen date.

DAYS BETWEEN DATES

At times you may want to know the number of days between two dates so that you can assess investment returns during a certain period. Rather than getting a calendar and counting, you can use the program to do this automatically. Just enter the two dates and press the Enter key.

GENERAL MARKET WISDOM

We have compiled some general theories on the real meaning of the market indicators contained in this program. You should use these theories only as guidelines in making your own judgments. Be aware that the market is tricky and may not always follow the anticipated path.

Days in which up volume exceeds down volume by 10 to 1 signal the beginnings of important upward moves. Days in which advances exceed declines by more than a 4-to-1 ratio are similar, if less powerful, harbingers of coming strength. (The converse of these may be equally true as harbingers of a bear romp.) An individual daily TRIN of .50 or less makes it very likely that the market will rise on the following day as well. A 10-day TRIN of .80 suggests an overbought market; a 1.20 figure, an oversold one. When the number of unchanged issues is growing, risk is growing as well. Premiums on index futures expand at tops, and decline (even go to discounts) at bottoms. A put/call ratio of 1 or more (i.e., where the number of put contracts traded equals or exceeds the number of calls) suggests a "buying opportunity". When the Dow average moves to new highs, and the broader averages and the advance/decline line fail to follow it, a top may be forming. A widening spread between the Fed funds rate and the discount rate, and situations in which short-term rates exceed long-term rates are bearish. Widening spreads between commercial paper and T-bill rates suggest fear of disaster, and can presage a crisis or a bottom. And remember the old Wall Street saying: "The young man knows the rules; the old man, the exceptions".

EXITING

You can easily exit to DOS from the Main Menu. You can return to the previous menu from most points in the program by pressing the Escape key.

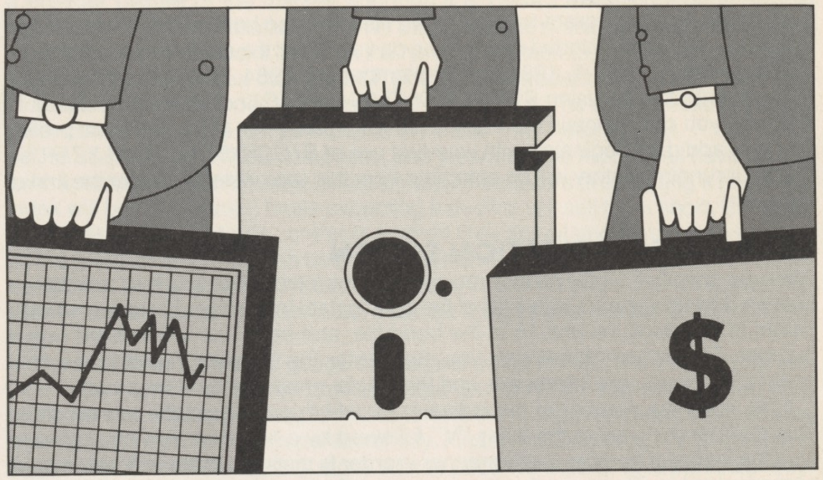
Remember to back up your data files on a separate data disk. Since the files will become large and complex as you use them, it will be very difficult to reconstruct your data if a problem occurs.

DATA SOURCE

The majority of stock and bond data used in this program was supplied to us by Hale Systems Inc., an online financial information service. If you have any questions regarding the information or would like to subscribe to the Hale Systems service, contact:

Michael Melzer
Product Manager
Hale Systems Inc.
1044 Northern Blvd.
Roslyn, NY 11576
(800) 645-3120 or (212) 895-3810

BUSINESS



STOCK/BOND DATA FILE

By Doug Henwood
Programmed by Evelyn Leon

Special Requirements: None
Files Used: SBDAT.BAS
STOCK.DAT
BOND.DAT

The STOCK/BOND DATA FILE *was designed to be used in conjunction with* STOCK/BOND ANALYZER. *However, this is a versatile data file which is also able to stand alone. Simply call up and view data from any of the 29 stock or bond indicators. Retrieve information for the current year up until September, 1984, and add your own data thereafter.*

START-UP

To gain access to the *STOCK/BOND DATA FILE*, you should first copy the necessary files to your working diskette. To make the copy, your system must be in DOS. Put your *PC Disk* diskette in the default drive and type:

```
COPY SBDAT.BAS B: ↵
```

where "B:" is the drive containing your working diskette. Then repeat this process to copy the other files, BOND.DAT and STOCK.DAT.

To display the information in the *STOCK/BOND DATA FILE*, insert a system disk in the default drive and load Advanced BASIC by typing:

```
BASICA ↵
```

Then put your working copy of the *STOCK/BOND* system in the default drive and type:

```
RUN "SBDAT ↵
```

The Title Screen appears first, followed by the three-option Main Menu. Enter "S" to view stock data, "B" to view bond data, or press the Escape key to exit the program and return to DOS.

The data in this program resides in two files that can hold information for up to ten years. We have included information on values for the stock and bond indicators between January 1, 1984 and September 30, 1984. You may add data for subsequent dates by using *STOCK/BOND ANALYZER* (included in this issue). In this way, you can maintain a complete running history for 1984. Or, if you prefer, you may add data beginning with your first use of *STOCK/BOND ANALYZER*, so that your running history will be complete from this date instead of from the first of the year.

THE DATA SPECIFICATION SCREEN

After you specify the type of data you wish to view (stock or bond), the Data Specification Screen appears with a list of the appropriate indicators. There are 14 indicators in the stock file and 15 in the bond file, and each one is identified with a number. To display information, you must enter the indicator number, the time span in which you are interested, and the display mode (screen or printer).

First, enter the number of the indicator for which you would like to view data. This number consists of the letter "V" followed by a numeral. You must always type the indicator number in this manner in order to distinguish the indicator from a numerical value. There are three ways to specify the indicator. If you wish to see all of the data for an indicator for a certain period of time, then respond to the prompt by typing:

```
Vn
```

where "n" is the number corresponding to the indicator of your choice.

The second method of specifying an indicator is useful if you wish to display data for a particular value. You may display the dates on which the value of the indicator was equal to, greater than, or less than a particular number. For example, to see the dates on which the Dow Jones Industrial Average was less than 1000, you would enter:

```
V1 < 1000 ↵
```

where "V1" is the indicator number, "<" is an operator, and "1000" is the comparison value.

The third method allows you to compare the values of two stock indicators or two bond indicators. The command:

```
Vx = Vy ↵
```

calls up those dates on which the first indicator (Vx) had a value equal to the second indicator (Vy). You could also use the operators for greater than (>) or less than (<).

Once you have entered an expression for the indicator, you must select the time frame for which you want data displayed. The program prompts you to enter the start date and the end date in the form MM/DD/YY.

Finally, specify whether you wish the data to be displayed on the screen or sent to a printer by typing either "S" or "P". Remember to check the printer to see if it is on.

STOCK INDICATORS

There are 15 indicators for stocks in *STOCK/BOND DATA FILE*. The first of these is the Dow Jones Industrial Average (DJIA), often referred to as the Dow. It represents the closing prices of 30 of the highest-quality blue-chip stocks.

The Standard & Poor's 500 is an index based on the performance of 500 blue-chip stocks. Since it is computed from a group much larger than the Dow, it gives a broader picture of the market, although it is biased toward the large, visible stocks preferred by institutions. The NYSE composite is an index representing the closing prices of all stocks listed on the New York Stock Exchange, which makes it an even broader index than the S&P 500. The broadest index of all is the Value Line index, which includes stocks from all exchanges. The smallest start-up computer company counts just as much in the Value Line index as IBM.

The S&P 500, the NYSE composite, and the Value line indexes all have futures contracts traded on their value. Most simply, speculators anticipating a rise in the index will buy the futures; those expecting a decline will sell them short. But when the prices of these futures contracts exceed their actual underlying index values by too great an amount, it can be a sign that the market is overbought, and due for a pullback. Similarly, when the futures trade at little or no premium—or even a discount—to their actual or cash value, it can be a sign that a market bottom is forming.

The next three indicators gauge the volume of the NYSE. The Total NYSE volume is the number of shares of stock listed on the NYSE that were traded on a particular day. The Advancing NYSE volume is the number of traded stock shares whose prices increased on a particular day. Finally, the Declining NYSE volume is the number of traded shares of stock whose price went down on a particular day.

Three more of the indicators deal with NYSE issues. The first, NYSE issues advancing, is the number of companies on the NYSE for which the stock prices increased. NYSE issues declining is the number of companies on the exchange whose stock prices decreased. And, as the name implies, NYSE issues unchanged is the number of companies whose prices remained the same.

The two last stock indicators are based on the Chicago Board Options Exchange. They measure the total number of call and put contracts traded on the CBOE for a given day. In general—though these things can be very tricky—if too many calls are traded, the market is probably overbought, and due for a pullback; if too many puts are traded, the market could well be due for a rally.

Index future premiums and put/call ratios are sentiment indicators, although their interpretation may sound wrong at first. If there are too many optimists, there aren't enough new buyers around to bring higher prices; if there are a lot of pessimists, all the selling is probably done, and the market can only turn up from here.

BOND INDICATORS

The indicator list for bonds in the *STOCK/BOND DATA FILE* numbers 15. The first indicator is the Discount rate. This is the rate that the Federal Reserve Bank charges member banks when it lends them money.

The second indicator is the Federal Funds rate. This is the rate that banks charge when they lend money to each other.

The third indicator is the commercial paper rate. This is the rate large corporations must pay for short-term borrowing.

The next indicators in the model are concerned with interest rates on U.S. Treasury securities, ranging from 3-month Treasury bills to 30-year Treasury bonds. Most other interest rates move up and down with Treasury rates; Treasury rates thus serve as a benchmark for the rest of the credit markets.

The relation between short- and long-term interest rates can offer a clue to future trends. If short-term rates exceed long-term rates, interest rates in general may rise significantly; conversely, short-term rates significantly lower than long-term rates can presage a decline in overall rates.

The next indicator, the Merrill Lynch Long Municipal bond rate, indicates the average yield on municipal bonds.

The next indicator on the bond list is the M1 money supply. This calculation by the Federal Reserve represents the total of all currency circulation plus checking account deposits. It is widely thought to represent a leading indicator of interest rates and economic activity. Thus, growth in M1 is correlated with lower interest rates, more vigorous economic activity, and rising inflation; when the money supply is flat or declining, interest rates are likely to rise, the economy is likely to contract, and inflation will decline.

The net free reserves, also compiled by the Federal Reserve, is the surplus or deficit in bank reserves relative to the amount the Fed requires them to keep. When free reserves are plentiful, the Fed is being easy; when free reserves are low or negative, the Fed is tightening.

DISPLAYING THE DATA

Although there are three ways to specify the type of data to be displayed, there are actually four different display screens. If your choice contains more data than can fit on one screen, you may scroll through the data by pressing:

6 →

to move forward, or pressing:

4 ←

to move backward.

The first of these screens appears when you elect to display all the values for an indicator during a specified interval. This screen shows the indicator number, the indicator name, a list of the dates, and, beside each date, the corresponding value.

The second screen appears when you compare an indicator to a particular value. This display includes the indicator number and name, the expression you are displaying, and the list of dates and values corresponding to the expression.

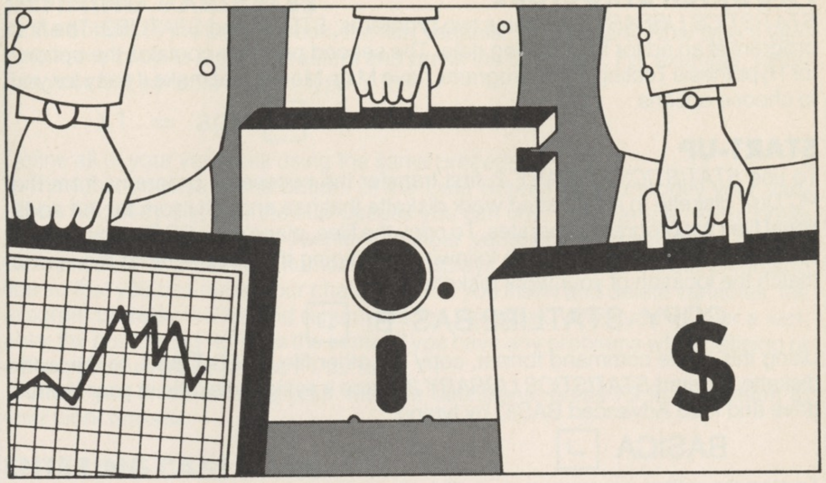
If the expression you entered requires the dates on which two indicators had equal values, the third possible display screen will appear. This screen shows both indicator numbers and names, the comparison expression, and the dates and values at which they were equal.

The fourth display screen lets you see the dates on which the value of one indicator was either greater than or less than the value of another indicator. This screen displays the number and name of each indicator and then shows three columns of information. The first column shows each date that fits the expression, the second lists the values of the first indicator for each date, and the last column does the same for the second indicator.

DATA SOURCE

The majority of stock and bond data used in this program was supplied to us by Hale Systems Inc., an online financial information service. If you have any questions regarding the information or would like to subscribe to the Hale Systems service, contact:

Michael Melzer
Product Manager
Hale Systems, Inc.
1044 Northern Blvd.
Roslyn, NY 11576
(800) 645-3120 or (212) 895-3810



STATISTICS LIBRARY 2

Designed by Dale Benzer
Programmed by Peter Schlaifer

Special Requirements: Min. 128K RAM (Printer optional)
Files Used: STATLIB2.BAS
STEDIT.BAS

Analyzing data is a tedious task which few people enjoy. With the help of STATISTICS LIBRARY 2, *however, you can analyze facts, test theories, and discover relationships without the strain of manual computations.* STATISTICS LIBRARY 2 *is the latest in a series of* PC Disk *statistical packages. Using this program, you can quickly and easily perform statistical analyses to help you make decisions based on information you've compiled.*

Anyone who wants to learn from the past and project the future will find this program an invaluable tool. For example, if you're interested in comparing one group of employees to other groups, this package can do the job. Or, it can help you to test your theories about the market place or to learn from stock/bond trends. You also have the ability to translate files from certain spreadsheet programs so that they can be used with the STATISTICS LIBRARY 2 *statistical routines. This is a real time-saving option for busy people.*

BACKGROUND

STATISTICS LIBRARY 2 lets you create data bases and test hypotheses. With the editor, you can define a list of up to 127 variables for statistical analysis and cross-tabulation. These variables can represent anything from survey responses to characteristics such as age, sex, and income. Once you have defined your variables, you may enter a different set of data for each variable for as many "cases" (individual questionnaires, etc.) as you have in your sample. Then, you can divide your variables into subsets you want to correlate (for example, age and income). Once your data base files are ready, you can test your theories about the data you've collected.

PROGRAM STRUCTURE

STATISTICS LIBRARY 2 contains two programs, STEDIT and STATLIB2. The first program is an editor for inputting data. The second program contains the options for Hypothesis Testing. Both programs have Main Menus that make it easy for you to choose options.

START-UP

To use *STATISTICS LIBRARY 2*, first transfer the necessary programs from the *PC Disk* diskette to a formatted work diskette that has enough room for the additional files the program generates. To copy the files, place your *PC Disk* diskette in the default drive. Then type the following, changing the drive letter at the end to match the location of your work diskette:

```
COPY STATLIB2.BAS B: ↵
```

Using this same command format, copy the other file, STEDIT.BAS, to the work diskette. To start *STATISTICS LIBRARY 2*, place a system diskette in your default drive and load Advanced BASIC by typing:

```
BASICA ↵
```

To start the editor, place your work diskette in the default drive and type:

```
RUN "STEDIT ↵
```

To perform statistical analyses, return to Advanced BASIC and type:

```
RUN "STATLIB2 ↵
```

THE EDITOR

To use the statistical analysis program, you must enter your data into special data base files which you either create using STEDIT or translate from spreadsheet programs using an option included with the editor. If you use files created with the editor from the *STATISTICS LIBRARY 1* package (published in *PC Disk* Vol. 1, No. 8), you need not make any changes to them. Conversely, you can use any files you create using the editor contained in this package with the analysis routines contained in *STATISTICS LIBRARY 1*.

After you start the editor, a Main Menu appears with the data-entry options. At the bottom of the menu, you see the function keys you can use with the program. For more information about these function keys, press F1. Under the function key definitions is the command line, where you enter your option choices and data.

CREATING A NEW FILE

To create a new data file, press:

```
C
```

and specify a filename. This name can have a maximum of eight characters. Next, you can enter a file description of up to 25 characters. This description helps you remember special information about your file such as test location, etc. On the next screen, choose the number of variables you want to use in your file (from 4-127).

The Data Screen appears next, along with the filename and the case number, which the program calculates automatically. You're now ready to define your variables.

DEFINING VARIABLES

At the bottom of the Data Screen, the first variable, x1, appears. Now type a variable name of up to eight characters and press the Enter key. For example, to define your first variable as age, type:

```
x1 = age ↵
```

Define all of your variables using the same procedure.

The function keys at the bottom of the Data Screen let you modify your variables. Each key displays the sub-options you can use. For example, if you press F4, you can delete, insert, reenter, or clear variables. Each of these options prompts you for a variable number and displays that variable on the command line so that you can make your change. When you insert and delete variables, the program updates the list and renumbers the variables. When you reenter a variable, the numbering remains the same. If you have any problems while defining or editing your variables, press F1 for help.

When you finish entering your variable definitions, press F3 to save them to your work diskette.

ENTERING CASE DATA

STATISTICS LIBRARY 2 only accepts data in numerical form, up to a maximum of four digits, so you may need to code your information before entering it. For some variables, such as age, you must specify ranges and code each range with a numeric value. For example, "1" can designate anyone less than 20 years old; "2" can designate anyone between 20 and 25, etc. For variables such as sex, you can simply use "1" for males and "2" for females. Avoid using the value of zero as a valid value.

When you're ready to enter information, press F2 to return to the Main Menu. Then, press:

```
I
```

to input case data. The variables you already defined appear on the screen along with the filename and case number. Variable x1 appears on the command line waiting for you to enter data. Type your data for the first variable and press the Enter key. The next variable, x2, will then appear on the command line. Continue entering data for all of your variables. If you ever need to change an entry, press F4 and choose option "R" to reenter data. Specify the variable you want to change, and enter the new data value.

After you finish entering data for a case, the program processes the information before you can enter data for the next case. It also adjusts the "Case x of x" line, in the upper right corner of the screen, as you finish each case.

Once you've entered data for all your cases, press F3 to save the data and return to the Main Menu.

DEFINING SUBSETS

You designate the variables you want to analyze by defining subsets. For example, if you're going to test a hypothesis on how people under 20 compare with the rest of a particular sample, then designate age group "1" as a subset.

To define a subset, press:

S

on the STEDIT Main Menu. The program prompts you for the definition of the first subset, s1. To define this subset for people under 20, first tell the program which variable contains the information (for example, variable 1), and then enter the numeric value you want to use. For example type x1 = 1, and press the Enter key. Next, you could define s2 for people between 20 and 25 by typing x1 = 2. While defining your subsets, you can insert, delete, or reenter information by pressing F4. Follow the steps outlined in the "Defining Variables" section.

THE REJECT CASE OPTION

In addition to rejecting cases when a subset is not true, *STATISTICS LIBRARY 2* lets you reject cases in which any variable has a value of zero (0). You can use this option to eliminate incomplete cases. Select the Reject option by typing:

R

before the subset definition. The program displays "REJIF0" as an entry on the Subset Screen.

THE RELATIONS <, >, <=, >=

Besides the relation for equality or identity (=), you can use these other logical relations when defining subsets: "<" for "less than", ">" for "greater than", "<=" for "less than or equal to", and ">=" for "greater than or equal to". If, for example, you want to identify all those people in your data base who answered question 4 on a survey with a 2 or above, you can define subset s4 as x4 (the variable for the question) >= 2. Using this subset, you can find responses that are not only equal to 2 but also greater than 2.

DERIVED VARIABLES

The derived variable option lets you define a new variable in terms of existing variables or constants. For example, you can set up a new variable, x4, to hold the values of variables x1 + x2. Then, you can perform all operations, including subset definitions, using this new variable. To define a derived variable, return to the STEDIT Main Menu and press:

D

Follow the same steps you used to enter a subset, but use the variable label "d" rather than "s". The program includes these new variables in the data file.

TRANSLATING .DIF FILES

If you have files created with a spreadsheet program and saved with the extension ".DIF", you can translate these files to work with the *STATISTICS LIBRARY* programs. Press:

T

on the STEDIT Main Menu. You will be asked for the .DIF filename and drive. Next, state whether the .DIF file was saved by row or by column. Then, choose a filename for the translated file (without an extension), and specify the drive you want to use.

The message, "Translating to .STL format", appears on your screen. When the program finishes translating your file, you will return to the Main Menu. You can then perform statistical analyses on your spreadsheet file using STATLIB1 or STATLIB2.

EXITING STEDIT.BAS

To exit the editor, press:

E

from the Main Menu. Once you return to BASIC, you can run the statistical routines program, STATLIB2.

STATISTICAL ANALYSIS

To perform statistical analysis on your data base files, start the STATLIB2 program by following the directions in the "Start-Up" section. The Main Menu lets you load a file into memory, select subsets, and test theories about the data you've stored (Hypothesis Testing).

LOADING FILES

Before you begin your statistical analysis, you first need to load a file into memory and select subsets. Press:

L

to load a file; type the filename, and specify the drive. After the file is loaded, the Main Menu appears again.

SELECTING SUBSETS

When analyzing your data, you can limit the case data under consideration by using more than one subset definition. Press:

S

to select subsets. Let's say you want to find the number of males over the age of 25 living in urban areas. To do so, you need to define three subsets—males, people over the age of 25, and people living in urban areas. Your definitions are as follows:

s1:x1>=3 (x1 is the variable for age; 3 is the number for people between the ages of 25 and 30.)
s2:x2=1 (x2 is the variable for sex; 1 is the number identifying males.)
s3:x3=1 (x3 is the variable for location; 3 is the code for urban areas.)

By choosing this combination of subsets, you can analyze data pertaining to this particular sample group.

HYPOTHESIS TESTING

The Hypothesis Testing option lets you perform tests on means or variances. First load a file into memory and select subsets. Then, press:

H

to test your hypothesis. A menu lists the four types of tests you can perform.

Let's try an example. Suppose a personnel department has given an intelligence test to all employees for several years. For all employees tested, the mean score equals 50 with a standard deviation of 10. Now, assume that you feel managers score higher on this test. To test this hypothesis, select managers as a subset, and then choose option 1 on the Hypothesis Testing Menu, Test of One Mean. The program asks you for the following information: the index on the variable to test (in this case, the variable for intelligence test scores, say x4), the hypothesized mean (in this case, 50), the standard deviation (10), and the confidence interval (between 70 and 99; let's say 99). In this case, the confidence interval is a range in which the sample mean (50) would fall 99% of the time if we were to do repeated sampling.

The program creates "null" hypotheses that assume the opposite of what you really are hoping to find. In this case, you're hoping managers score above 50, so the program forms null hypotheses stating that the managers' mean is equal in some way to 50. If these hypotheses are rejected, then we can accept that managers' scores did not equal 50.

The sample mean is the average of all the scores. The sample standard deviation is a measure of variation around the sample mean. The sample variance is the square of the standard deviation of the sample. The standard error is the standard deviation divided by the square root of the sample size. In this program, a "normal" distribution is used when you know the standard deviation. If you don't know the standard deviation, press the Enter key, and the program will choose the Student's *t* as the test statistic. Next to the word "normal", you're told the *z* score for the distribution. The *z* score is the number of standard deviations your statistics stray from the mean. The confidence level indicates the confidence limits placed on your mean.

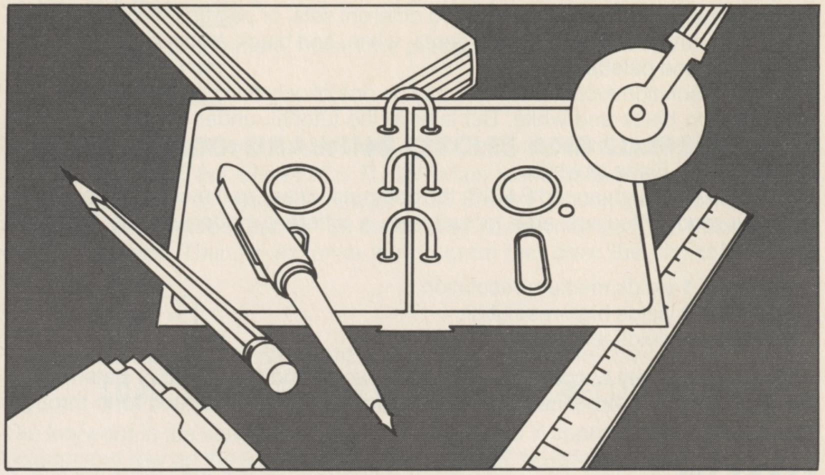
At the bottom of the screen, you see the possible null hypotheses developed to test your theory. As we mentioned, null hypotheses are educated guesses created with the hope that they will be rejected. Since you're hoping that managers score higher than 50, the null hypotheses are based on the theory that the scores are somehow equal to 50. Next to the null hypotheses are the alternative hypotheses—those other than the null hypotheses. You are hoping to accept one of the alternative hypotheses by rejecting the associated null hypothesis.

The "significance level" at the right determines whether you should accept or reject the null hypothesis based on the probability of committing a Type I error (rejecting a hypothesis that should be accepted). The higher the significance level, the more likely it is that you will commit an error if you reject the null hypothesis. Conversely, the lower the number, the stronger the evidence that you should reject the null hypothesis and accept the alternative.

EXITING STATLIB2.BAS

To exit STATLIB2.BAS, return to the Main Menu and choose option "E".

EDUCATION



GRAPHICS TUTOR

By Stan W. Merrill

Special Requirements: Color/Graphics Adapter
Files Used: TUTOR3.BAS

The IBM PC has several display modes. In our program entitled BASIC BOOTSTRAPS *contained in Vol. 1, No. 8, we discussed character graphics, which are especially useful for creating forms and charts. Now we'll turn to two of the PC's graphics modes: medium-resolution and high-resolution graphics.*

Medium- and high-resolution graphics (often called med-res and hi-res for short) provide the ability to create detailed pictures. In this tutorial, we'll make line, bar, and pie graphs as well as some "art" using circles, boxes, lines, points, and a special drawing language.

To use medium-resolution and high-resolution graphics, you need special equipment: a color/graphics adapter card and a monitor that will attach to the card. If you don't have a color/graphics adapter, you will be unable to view this tutorial on the computer itself. But you may want to print the program out and study it as you read through the tutorial.

MEDIUM-RESOLUTION VERSUS HIGH-RESOLUTION

There are two main differences between medium- and high-resolution graphics. The first is the number of dots on the screen. In medium-resolution mode, there are 320 dots (labeled 0-319) across the screen and 200 dots (labeled 0-199) down the screen for a total of 64000 dots. Every dot on the screen may be specified as a pair of column and row coordinates starting with 0, 0 at the top left of the screen and ending with 319, 199 at the bottom right. For example, the center of the screen is 160, 100.

In high-resolution mode there are 640 dots across the screen instead of 320, making a total of 128000 dots. The column coordinates range from 0-639, while the row coordinates are still between 0 and 199 as with medium-resolution mode.

The second difference is the number of colors. In medium-resolution mode, a dot may be one of seven colors, although there are only four colors available at any one time. You have a choice of two different sets, or palettes, of color: green, red, brown, and black; or cyan, magenta, white, and black. You may use only the colors from one palette at a time.

In high-resolution mode, each dot may be one of only two colors. Officially, you are limited to black and white. But later in the tutorial, under the heading "Advanced Topic: Medium and Hi-Res Color", we'll explain how and why you can use other colors in place of white.

When you start Advanced BASIC, the computer uses "text mode" rather than a graphics mode. You are able to switch to a different mode with the SCREEN statement.

SCREEN 1 yields medium-resolution.
SCREEN 2 yields high-resolution.
SCREEN 0 yields text mode.

In the programs that accompany this tutorial, you'll find the SCREEN statement at the start of each subroutine because the program shifts back and forth through these modes.

START-UP

To start *GRAPHICS TUTOR*, insert a system disk in your default drive and load Advanced BASIC into your PC by typing:

```
BASICA ↵
```

Then insert your work copy of your *PC Disk* diskette into your default drive and type:

```
RUN "TUTOR3 ↵
```

The program begins with a representation of the ten function keys of the IBM PC. Each key is labeled with the graphics screen that will be displayed if you press that key.

After you experiment with the graphics displays of *GRAPHICS TUTOR*, you will want to see the programming code used to create them. There are two ways of doing this: you can either display the code directly on the screen or print out the entire program for easy reference. To view the code on your monitor, press the Alt key and the desired function key simultaneously. For example, to view the code used to generate the rubber bands display, simultaneously press:

Alt F1

Doing so will return you to BASIC and display the code that is implemented when you press Function key 1. If the code scrolls off the top of the screen too quickly, press:

Ctrl Num Lock

to freeze the display. Resume scrolling by pressing any key.

To resume the program and continue experimenting with the other graphics features, type:

```
RUN ↵
```

If you would like to have a hard copy of the program, exit to Advanced BASIC. Then make sure your printer is ready and type:

```
LLIST ↵
```

When the listing is complete, restart the program by typing:

```
RUN ↵
```

RUBBER BANDS: DRAWING CIRCLES AND ELLIPSES

The first option on the *GRAPHICS TUTOR* Main Menu uses Advanced BASIC's CIRCLE statement to draw a variety of arcs. Press Function key 1 to watch the computer draw "rubber bands" on the screen. To return to the Main Menu, press the Escape Key. Then, to examine the program that drew the circles, simultaneously press:

Alt F1

You will find the CIRCLE statement in line 1940. It looks like this:

```
1940 CIRCLE (X,Y),R,TINT,,,ASPECT.RATIO
```

The keyword in the statement is CIRCLE. The X and Y coordinates represent the column and row on the screen where the center of the circle will be.

The "R" in line 1940 stands for radius, or the distance from the center to the outside of the circle.

The TINT, or color, of the circle is a number from 0 to 3 in medium-resolution mode or either 0 or 1 in high-resolution mode.

The last item in line 1940, the aspect ratio, is necessary because the screen is wider than it is tall. Since every point on a circle should be the same distance from the center, the computer must adjust the number of dots it uses depending on the direction in which it is drawing. The aspect ratio parameter lets you specify the relationship between horizontal and vertical distances on the screen. The default values for the aspect ratio are 5/6 in medium-resolution and 5/12 in high-resolution. These values can be used to display circles on a standard screen.

You can also use aspect ratios to make ellipses. Aspect ratios less than one will stretch the circle in the horizontal direction, while ratios larger than one will stretch it in the vertical direction. To draw a tall, narrow ellipse, use an aspect ratio of 2 or 3, and use an aspect ratio of 3/16 or 1/25 to draw a wide, squat ellipse. When you make ellipses, the radius you give will be the length of the long axis. The computer will automatically shorten the radius as it moves away from the long axis toward the short one.

Now you can easily understand the program that produces the rubber bands. Lines 1880-1920 each produce a parameter (X, Y, radius, tint, aspect ratio) by randomly generating a number within the required range. Then the program calls the CIRCLE statement with the parameters.

PIE CHART: DRAWING ARCS AND PAINTING SHAPES

You can display a helpful business tool, the pie chart, by pressing Function key 2. A pie chart presents figures in an understandable form by dividing a circle into wedges, each of which occupies a proportion of the total.

The CIRCLE statement provides a method of dividing a circle into wedges so we can make pie charts. Now that you've seen an example, press the Escape key, then simultaneously press:

Alt F2

to look at the program code used to create the pie chart.

In line 3240, you will see the familiar CIRCLE statement, with the X and Y coordinates, radius, color, and aspect ratio. But now there are two more parameters: the start point and the end point. When these parameters are included, the computer draws only a portion of a circle or arc, extending from the start point to the end point.

The start and end points are angles, given in radians. There are 2 times pi, or about 6.28, radians in a circle, so the start and end points will lie between 0 and roughly 6.28. One radian is the same as about 57 degrees.

To make a pie chart, first add the numbers that will be included, then divide each one by the total to find out what portion of the total it is.

Let's assume a starting point of zero. Then we can find the end point of the arc for a particular data item by multiplying the number of radians in a circle by the proportion of the total which that data item contributes. For succeeding arcs, we use the starting point as the end point of the last arc and calculate a new end point. This is done in lines 3210-3240 of the program.

To make it easier to judge the relative sizes of each wedge, it is customary to color them in. In Advanced BASIC, we color objects with the PAINT statement, which looks like this:

```
PAINT (X,Y),paint,boundary
```

where "(X,Y)" are the coordinates of a point inside the area to be painted. The area can be any shape, and the point indicated can be any point within it.

The paint is the color you want the inside to be and the boundary is the color of the lines that encompass the area. The PAINT statement will paint the entire area with the paint color until it reaches the boundary color. To fill an area in, it must be completely surrounded by the boundary color. Otherwise, the paint will spill out of the area and continue painting other parts of the screen. The colors for paint and boundary can vary between 0 and 3 in medium-resolution and 0 and 1 in high-resolution.

In the pie chart program, we must find a point within a wedge and determine its X and Y coordinates. Lines 3260-3280 do this by calculating the middle of the wedge and translating the coordinates from radians to X, Y points. Then line 3300 colors the wedge using the PAINT statement.

Finally, lines 3330-3370 label each wedge by calculating a point completely outside the circle and printing a label at that point.

The labels for the pie chart (as well as for the line graph and bar chart also included in the program) are stored in lines 180-190 of the program. The data is found in lines 160-170. You may replace the existing data and the labels to make your own charts.

LINE GRAPH: DRAWING STRAIGHT LINES

Another familiar type of chart is the line graph. To look at an example, press Function key 3. This line graph uses the same data and labels as the pie chart. When you've seen what the program does, look at the code by pressing the Escape key, then simultaneously pressing:

Alt F3

As you can see, the program first reads the data, then it calls subroutines to draw the scale axes and label them. The LINE statement that draws the axes is found in line 2350.

The coordinates in the first set of parentheses specify the starting point of the line, and the second pair specifies the end point. This particular statement draws a vertical line on the screen if the X coordinates are identical. If the Y coordinates were identical and the X coordinates were different, it would draw a horizontal line. If all the coordinates were different, it would draw a diagonal line.

In lines 2070-2110, the program plots the X and Y coordinates that represent the data values. It saves the coordinates in two arrays, X() and Y(), for later use. Then it draws a small circle, using the CIRCLE statement, around each plotted data point.

Lines 2130-2150 extract the X and Y coordinates from the arrays and draw a line connecting each pair of them.

BAR CHART: MAKING BOXES

An alternative to the line graph is the bar chart. You can see an example of one by pressing Function key 4. This bar chart uses the same data and labels as the pie chart and line graph. To see how it was created, press the Escape key, then simultaneously press:

Alt F4

As with the line graph, the program reads the data and calls subroutines to draw the scale axes and label them. But instead of calculating points and drawing lines through them, it uses another ability of the LINE statement: drawing boxes. To draw boxes with the LINE statement, enter the top left-hand coordinates of a box and the bottom right-hand coordinates, then add a letter "B" at the end of the statement. If you also add an "F" after the "B", the program will fill the box with the same color as the outline.

Lines 2960-2990 of the program calculate the top and bottom corners of a box. Once the coordinates are calculated, the LINE statement in line 3000 draws a box and fills it in.

TURTLE: SAVING PICTURES

Let's take a different approach to this section and look at the program code before we see what it does. This program provides some rudimentary "turtle" graphics. That is, you use it to draw on the screen much as you would use a pencil. To see the program code, simultaneously press:

Alt F5

The TURTLE program waits for you to press the cursor control (arrow) keys on the numeric keypad. When you press one of these keys, a trail of dots appears on the screen. The Left-Arrow key will move the dot left, the Up-Arrow key will move it up, and so on, just as the arrows on the keypad indicate. The dot will also move diagonally when you press:

1 End 3 PgnDn 7 Home 9 PgUp

Pressing the numeral 5 key will return the cursor to the center of the screen.

Line 3480 toggles the Num(eral) Lock feature so that the keys will always act like number keys to the program. But instead of printing numbers on the screen, the program uses the numbers to determine the direction in which to move the coordinates. Then the PSET statement in line 3710 moves the cursor to the appropriate place and puts a point there with the desired tint.

Notice that lines 3570-3600 read your key entry, then increment or decrement the X and Y coordinates accordingly. Line 3620 moves the point back to the center when you press the numeral 5 key. Line 3700 erases the screen when you press the Delete key. To change the color of the dot (line 3680), press:

0 Ins

The dot starts as green, but changes to red, then to brown, and finally to black. Since the background color of the screen is also black, you don't see the dot while it's black. You can use a black dot to erase dots drawn on the screen in other colors. You can also use black to move from one spot on the screen to another without leaving a trail.

The routine for saving the picture you've drawn is found in line 3640. While you draw the picture, it is kept in the computer's graphics memory, starting at address &hB800. There is an Advanced BASIC statement, BSAVE, that stores a copy of memory on disk and another statement, BLOAD, that retrieves a copy from disk and places it back into memory. By telling the BSAVE command to save graphics memory, you can save an exact copy of any picture you draw.

The BSAVE command looks like this:

```
BSAVE filename, start.address, number.of.bytes
```

Before using this command, we must point to graphics memory using the DEF SEG statement as shown in line 3640 of the example. Then we must save memory starting at byte 0 and going up to 16 kilobytes, because graphics memory requires that much space.

The BLOAD command loads a memory-image file (your graphics image) created by BSAVE into memory, erasing whatever was there before. Again, we must point to graphics memory with the DEF SEG statement and then specify the name of the file to load.

Now that you understand how the program works, you can use it to create your own pictures. Press Function key 5. With a little practice, you can draw almost anything. Remember to press the Escape key to exit the TURTLE program.

NEON GIRL: USING THE DRAW GRAPHICS LANGUAGE

Some pictures are so complex that using circles and lines would be tedious and would require many lines of programming. In such cases, the DRAW statement may be a better choice. Press Function key 6 to display a picture of a girl made with the DRAW statement. Next press the Escape key, then simultaneously press:

Alt F6

The DRAW statement in Advanced BASIC is a small language of its own with many commands. The neon girl was drawn using just 11 of them.

A string of characters in the DRAW language instructs the computer to move the cursor up or down or in other directions, just as if you were pressing the cursor control keys while using the TURTLE subroutine. In most cases, a command in the string is a letter that tells the direction to move, followed by a number telling how many dots to move. The directional commands are:

L	Left
R	Right
U	Up
D	Down
E	Right and Up
F	Right and Down
G	Left and Down
H	Left and Up

For example, the string:

```
DRAW "U6R5D6L5"
```

will plot six dots up, five dots right, six dots down and five dots left, making a small rectangle. If you try to enter this example, first type:

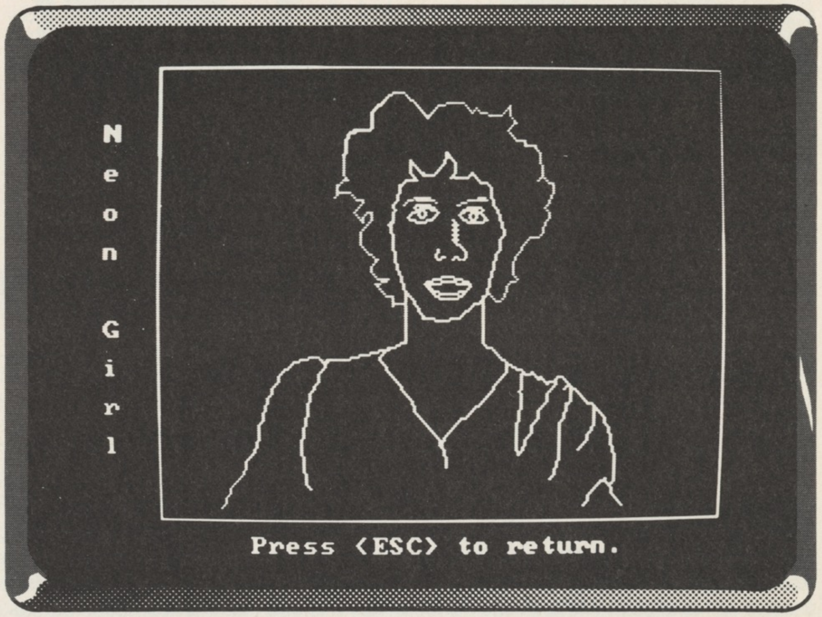
```
SCREEN 1 ↵
```

to enter medium-resolution graphics mode.

Prefacing any command with the letter "B" will cause the cursor to move without drawing. Thus:

```
DRAW "BU6BR5BD6BL5"
```

will move the cursor around the outline of the same rectangle, but will not actually draw anything. Thus, adding a "B" in front of any command lets you move to a new place on the screen without leaving a trail.



Press <ESC> to return.

Flashy graphics are a snap

The "M" (for MOVE) command differs somewhat from the others because it requires two numbers after it instead of one. The numbers are X (column) and Y (row) coordinates, separated by a comma. The MOVE command moves the cursor directly to the point on the screen specified by the coordinates. Again, adding a "B" in front of the MOVE command lets you move to a new place on the screen without leaving a trail. For example:

```
BM160,100
```

will move the cursor to the center of the screen in medium- resolution mode without drawing a line from its previous location.

Finally, the "C" (COLOR) command lets you change colors. In medium-resolution mode, you may choose any of four colors by specifying a number from 0-3. In high-resolution, you may choose 0 or 1.

MEDIUM- AND HIGH-RESOLUTION COLOR

As mentioned previously, there are two "palettes" of medium-resolution color. The first palette consists of green, red, and brown. The second includes cyan, magenta, and white. You may use either palette, but not both at the same time. Depending on the palette you choose, color 1 will be cyan or green, color 2 will be magenta or red, and 3 will be white or brown.

The COLOR statement selects the palette. In the graphics modes, its syntax is:

```
COLOR background, palette
```

The background is a color from 0-3, and the palette is 0 (green, red, brown) or 1 (cyan, magenta, white).

Lines 4620-4670 of the Neon Girl subroutine show how to use the palettes to achieve the neon effect. The program simply switches between palettes rapidly so that white alternates with brown, magenta with red, and cyan with green. This effect continues until you stop it. (Since the neon effect relies on color changes, you probably won't notice it if you don't use a color monitor.)

One of the unfortunate restrictions of Advanced BASIC is that text printed in medium-resolution is always color 3. So any time you PRINT something on the screen, it will be white or brown, depending on the palette you've chosen.

There is a way around this restriction. To see it, press Function key 7: The computer will print a message in all the colors in both palettes, although you won't see the message in black because the background is also black.

Now press the Escape key and then simultaneously press:

Alt F7

to look at the program. The code for this subroutine is straightforward. It selects the first palette, then prints the message in each of four tints. Then it selects the second palette and repeats the message. What causes the text to change tint is line 4030. This line POKEs the color number for foreground color into data location 78.

In high-resolution mode, there are 128000 dots on the screen. Dividing 16000 bytes of graphics memory into 128000 dots gives eight, indicating that eight dots must be stored in each byte. That leaves only one bit per dot. Since each bit can have the values 0 or 1, there are two possible values for each dot on the screen. According to the IBM manuals, a dot with a value of 0 is always black and one with a value of 1 is always white. Press Function key 8 to see that this restriction actually does not exist. The computer will draw a high-resolution geometric design, then will make it change colors. As you can see, while the design can be only one color at a time, it doesn't necessarily have to be white.

You can view the code that produced this effect by pressing the Escape key, then simultaneously pressing:

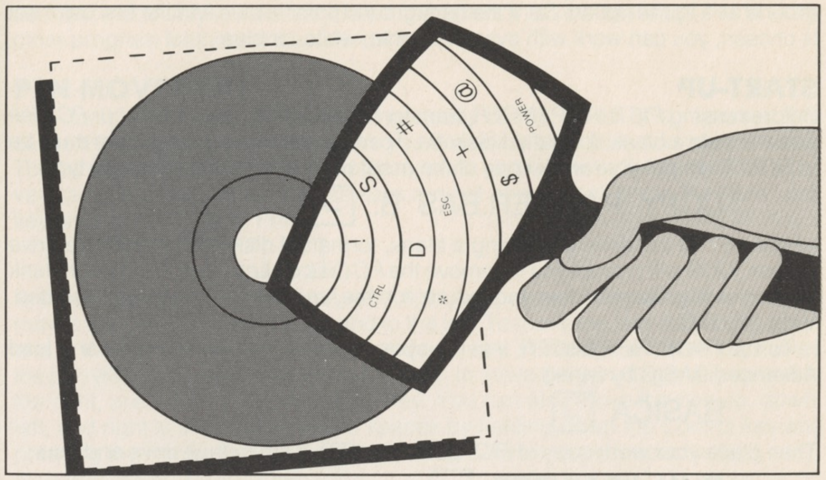
Alt F8

Lines 4780-4840 draw the geometric shape. Line 4870 sends a color value to port 985, causing the foreground color to change.

EXITING

To end *GRAPHICS TUTOR* from the Main Menu, press Function key 10 and the system will return to DOS.

UTILITIES



PICTURE PAINTER

By Michael B. Hildebrand

Special Requirements: Color/Graphics Adapter
Files Used: PAINTER.BAS
VALLEY.BAS

Create colorful, medium-resolution graphics quickly and easily with PICTURE PAINTER. *This pen-oriented graphics program lets you use Advanced BASIC's graphics capabilities to design, store, and retrieve screens.* PICTURE PAINTER's *simple commands facilitate the creative process. Draw lines, boxes, and circles automatically, or use the program's Draw mode to make your own shapes. You can even use color and text to supplement your artwork, and call up these screens in your other BASIC programs. Whether your artistic abilities rival those of Michaelangelo or the average kindergarten student, you can use* PICTURE PAINTER *to start your own gallery.*

BACKGROUND

There are two graphics modes available on the PC: medium-resolution and high-resolution. In these two modes, the resolution surpasses that of the standard text display mode because the screen consists of thousands of individual points or dots.

PICTURE PAINTER operates in the medium-resolution mode, so its screen is also designed in this way. There are 320 horizontal dots and 200 vertical dots, numbered from left to right and top to bottom, starting with zero. Therefore, the upper left-hand corner coordinates are 0,0, while the lower right-hand corner is located at 319,199.

This graphics mode has a unique color scheme in that the PC does not color the screen dots and the screen background in the same way. The background can be any one of 16 available colors. A list of these colors appears later in this article under the section entitled "Color Control". As for the screen dots that are displayed over this background, there are only two sets of colors, or palettes, available for each dot: 1) cyan, magenta, and white, and 2) green, red, and brown. You may use only one of these three colors at one time, or select the color of the background as a fourth option. So, if the background color is blue and the first color set is chosen, you can work with cyan, magenta, white, or blue.

START-UP

Before running *PICTURE PAINTER*, transfer the necessary files from your *PC Disk* diskette onto a blank, formatted diskette. To make the copy, your system must be in DOS. Then put your work copy of the program in the default drive and type:

```
COPY PAINTER.BAS B: ↵
```

where "B:" is the drive containing a blank, formatted diskette. On a single-drive system, DOS will prompt you to remove the *PC Disk* diskette and insert your blank work diskette. Repeat this procedure for the sample file we have included, VALLEY.BAS.

To start *PICTURE PAINTER*, insert a system disk in your default drive and load Advanced BASIC by typing:

```
BASICA ↵
```

Then place your work copy of *PICTURE PAINTER* in the default drive and type:

```
RUN "PAINTER ↵
```

After the Title Screen appears, press any key to begin. You will be asked whether you want help. If you do, press:

```
Y
```

The Instructions Screen describes the syntax of the commands available in *PICTURE PAINTER*. You may page through the instructions by pressing the Space Bar. After familiarizing yourself with the commands, press any key to continue. You may access the Instructions Screen at any time during program operation, except while in Text mode, and then press the Escape key to continue where you left off.

To get an idea of the screens you can create with *PICTURE PAINTER*, use the restore command and load the sample file by pressing "R". When prompted for a filename, type VALLEY.BAS, then press the Enter key. In future *PICTURE PAINTER* sessions, you may use the restore command to display any screen that was previously created. When you restore an old file, the picture appears on the screen and you may use *PICTURE PAINTER's* commands to edit it. When you have finished viewing the sample file, press "E" to exit and then resume to create a new screen display.

When you create a new file, the screen first clears and then the program draws a border around it. A dot, representing your pen, appears at the center of the screen. Now you can use the available commands to design your own graphics. The commands available in *PICTURE PAINTER* enable you to draw, color, edit, and save the graphic screens you create.

PROGRAM MODES

There are three modes that you can use to create forms with *PICTURE PAINTER*. When you first start up the program, you will be in the default mode. Here you can create shapes automatically and manipulate colors. The cursor dot is active but will not leave a trace as it is moved. In order to use this dot as a pen, you must use Draw mode; you can then draw any shape you wish. Although you cannot draw shapes automatically in the Draw mode, you can change colors. Finally, you will use Text mode only to insert keyboard characters into your drawings. None of the other program features are active in this mode.

PEN MOVEMENT

You don't have to worry about having steady hands, rulers or compasses, because this program makes it easy for you to draw straight lines and round circles. The cursor control (arrow) keys on the numeric keypad move the pen horizontally, vertically, and diagonally. If these keys are not operational, press the Num Lock key once and try again. Press:

[8 ↑] [2 ↓] [4 ←] [6 →]

to move the pen up, down, left, and right, respectively. Pressing the End key moves the pen diagonally down and to the left. To move the pen diagonally down and to the right, press the Pg Dn (page down) key. Pressing the Home key moves the pen diagonally up and to the left. Finally, to move the pen diagonally up and to the right, press the Pg Up (page up) key. You can also move the pen up, down, left, and right in increments of five spaces by holding down the Control key and pressing the Home, End, Left-Arrow and Right-Arrow keys, respectively.

COLOR CONTROL

PICTURE PAINTER lets you change the colors of the background, the pen, and the border. A simple command allows you to change the color of the background. Type:

```
BBxB
```

where "x" is a number from zero to 15. The following is a list of the possible background colors and the commands used to invoke them.

Black	**BB0B**
Blue	**BB1B**
Green	**BB2B**
Cyan	**BB3B**
Red	**BB4B**
Magenta	**BB5B**
Brown	**BB6B**
White	**BB7B**
Gray	**BB8B**
Light Blue	**BB9B**
Light Green	**BB10B**
Light Cyan	**BB11B**
Light Red	**BB12B**
Light Magenta	**BB13B**
Yellow	**BB14B**
High-Intensity White	**BB15B**

The color selections for the pen and the border are more limited; *PICTURE PAINTER* has two palettes, or sets of colors, available for both. The default palette consists of green, red, brown, and the present background color. To change the color of the pen to the color of the background or to green, red, or brown, press "0", "1", "2", or "3", respectively. Change the color of the border by typing "By", where "y" is a number between zero and three.

To toggle to the second palette, press:

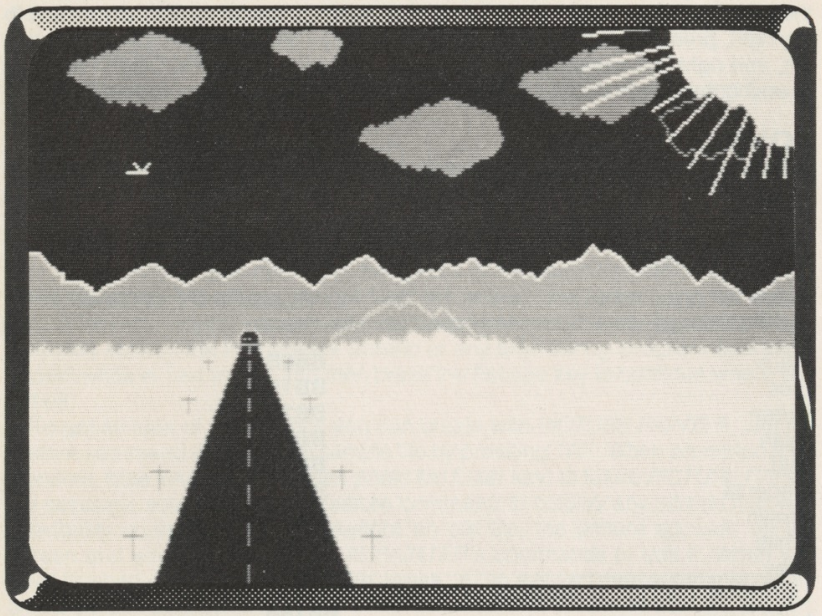
=

You may now change the pen to the current background color or to cyan, magenta, or white by pressing "0", "1", "2", or "3", respectively. If you cannot see the pen's current location, press "A" and the pen point will blink. Change the border color again by preceding one of these numbers with a "B". Return to the first palette at any time by pressing:

—

DRAWING COMMANDS

You can enter Draw mode at any time except while you are in Text mode (described below). To begin drawing, press "D". In this mode, all motions of the pen will be left on the screen as pen markings. The word "Draw" remains in the upper left corner of the screen to remind you that the program is operating in this mode. To exit from Draw mode, press the Escape key.



Pretty as a picture

If you make a minor mistake in one of your designs, you do not have to clear the screen and start over. Instead, you may erase a fraction of your design. First change the color of the pen to the background color by pressing "0". The dot representing the pen will disappear; you will not be able to distinguish it from the background. Now, while in Draw mode, use the cursor control arrows to overwrite the part of your screen you wish to erase. When you are finished, change the pen to a different color and continue with your design.

The program has a save feature that enables you to store your current screens before experimenting on them. If you don't like the results of your subsequent work, you can simply restore your original screen. To save a displayed screen, press:

X

To restore a saved screen, press:

U

PAINTING

As the program's name implies, *PICTURE PAINTER* allows you to paint segments of your graphics. To paint a particular closed figure, move the pen into the circle, box, or other shape and type the following command:

mPn

where "m" stands for the number of the color you wish the figure to be painted. If you are using the default palette and you wish to paint the object red, then this number would be "2". The second number, "n", is the number of the outline of the figure. For example, if you wish to color a box that has a green outline, then the second number would be "1" because that is the number for green in the default palette. So the command:

2P1

will take the object that is outlined in green and color it red. If you have toggled to the other color palette, this command will paint a cyan box magenta.

If, once you have painted an object, you decide you don't like it, press "U" to restore the screen to the color it was before the last "paint" operation.

AUTOMATIC SHAPES

Several of the commands in *PICTURE PAINTER* enable you to automatically draw lines, boxes, and circles. To create these figures, you must use the pen movement commands to mark locations on the screen, then use the automatic shape commands to generate the various forms. To mark the current position of the pen, press:

M

You may also use the "A" command after you have marked a spot. Pressing "A", in this situation, moves the pen from its present location to the last marked spot. The location of the pen when you pressed "A" will become the new marked spot.

You may easily draw a line between the pen's current position and the last position you marked by pressing:

LL

A similar command allows you to draw boxes. Position the cursor diagonally opposite the last spot you marked, then type:

LB

and a box will be drawn.

There are two equally simple ways to draw circles with *PICTURE PAINTER*. One method is to type:

CrC

where "r" is a number measuring the radius of your circle. The center of the circle will be the current location of the pen. Alternatively, draw a circle by typing:

```
CC
```

With this command, the radius will be the distance from the pen's present location to the last spot you marked. The current location of the pen will again be the center of the circle.

BLOCK COMMANDS

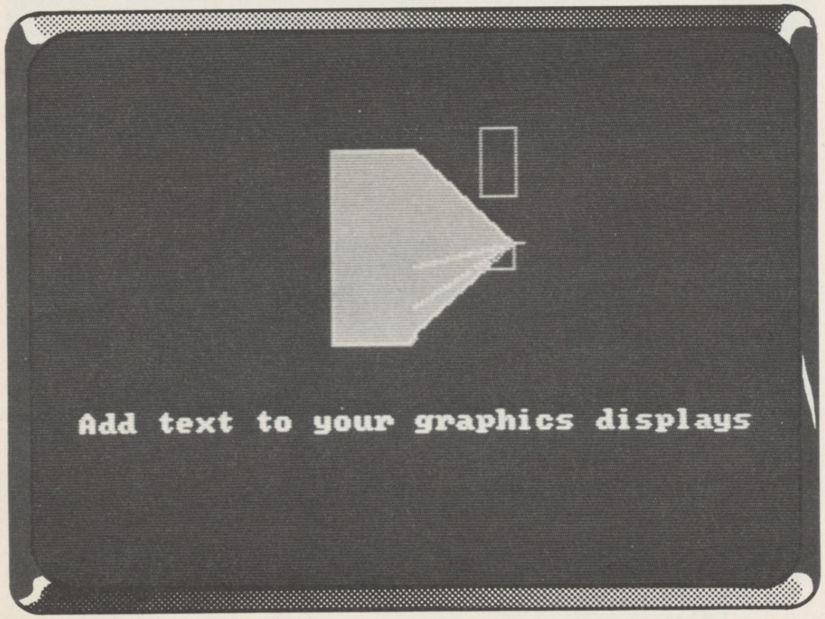
PICTURE PAINTER contains block commands that allow you to manipulate the screens you create. Before attempting the commands that allow you to save and copy blocks, you must first mark the block using the "M" command. Move the pen to the lower right corner of the block you wish to mark and press "M". Then, by repeatedly pressing:

7 Home

move the cursor to the upper left corner of the block. Finally, save this marked block by pressing "S". To show that this Save command was successful, the block will be momentarily highlighted. You may now copy this marked block to another part of your screen. First use the cursor control keys to move the pen to a new location. Then press "K" and the saved block will be copied. Copying a block in this manner works only if the saved block will not exceed the border when it is moved. If the block overlaps, move the pen to another position and try again. If you decide after moving the block that you liked it better in its original location, press:

```
U
```

and the screen will be restored to its status at the save command.



Add text to your graphics displays

Add text to your screens

TEXT MODE

If you wish to put letters, numbers, or special characters into your graphics, press:

```
T
```

to enter Text mode. The word "Text" will appear to let you know you are in this mode. You may now place any printable characters anywhere on the screen. Just use the cursor control keys to move the pen to the desired spot and then begin to type. If the character you wish to type is unavailable on your keyboard, hold down the Alt key and enter the ASCII value of this character. (To find the ASCII code for a particular character, consult Appendix G of the *IBM BASIC Manual*.) Please note that you may not enter any letter commands in this mode, because any letters pressed will appear as text on your screen.

If you make any mistakes, you can also erase text. First press the Escape key to exit from Text mode. Next, change the pen color to the background color by pressing "0". Enter Text mode again by pressing:

```
T
```

Then use the cursor control arrows to position the pen at the character to be erased. Press the Space Bar and the character will disappear.

USING YOUR DISPLAYS

Screens that you create with *PICTURE PAINTER* may simply be admired or they may be used in your BASIC programs. To use a screen, make sure it resides on the same disk as the BASIC program in which you will place it. Then insert a line in your program which will call up your screen. For instance, if you have created a title screen for a program and you have named the file for this screen FRONTEND.BAS, you would put a line in your BASIC program similar to the following:

```
100 BLOAD"FRONTEND
```

For further information about the BLOAD command, refer to Chapter 4 of the IBM BASIC Manual.

EXITING

To exit the program at any time except while in Text mode, press:

```
E
```

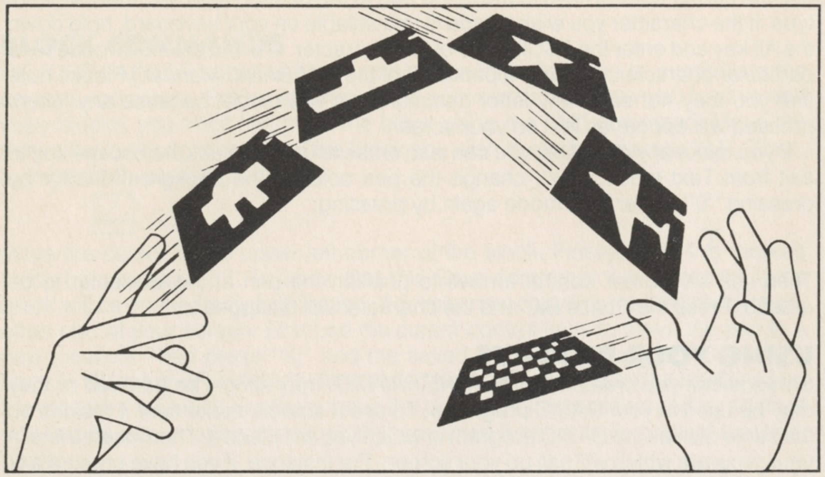
The program will first ask if you would like to save the display you have created. If so, you will be prompted to enter a filename. The program then asks if you wish to continue editing your display. You may continue to make changes with *PICTURE PAINTER* or exit the program and return to DOS.

SUBSCRIPTION INQUIRIES

If you have any questions regarding subscriptions, or you would like to renew your subscription or order back issues, call:

800-932-0017
in NJ 800-624-0497
or write:
PC Disk
P.O. Box 5930
Cherry Hill, NJ 08034

GAMES



TIC TAC GO

By Ron Dubren
Programmed by Peter Schlaifer

Special Requirements: None
Files Used: TTG.BAS

Everywhere we look, we must organize the chaotic visual world into recognizable and meaningful shapes. We see words, signs, symbols, and colors constantly, yet our minds make sense out of all this confusion. Although recognizing patterns and objects is unconscious, there can also be pleasure in discerning patterns in the visual world. TIC TAC GO *is a unique game of visual perception which tests your brain's ability to see and match patterns quickly.*

BACKGROUND

From infancy onward, we get better at "seeing" the world. First, we perceive simple shapes and learn the difference between squares, circles, etc. Gradually, certain patterns come to represent specific objects with depth and texture. Similarly, the acquisition of language transforms meaningless lines and curves into the alphabet. Before long, we think nothing of combining a series of squiggles into words on paper. *TIC TAC GO* is an enjoyable and challenging way to sharpen these pattern recognition skills.

GAME STRUCTURE

TIC TAC GO is played with a 3-by-3 grid, containing up to nine different patterns, and a target pattern. The object of the game is to select the grid pattern which exactly matches the target pattern. You must do this within a specified time interval, as indicated by a diminishing vertical bar on the right of the screen. You earn points for each correct match.

To make the game a little harder, you must accumulate a designated point score for each round in order to stay in the game. You have only three "lives" per round, or three allowances for incorrect answers. One life is lost whenever you take too much time to answer or you answer incorrectly. The game ends after you make four wrong guesses or you fail to reach the goal score for a specific round.

START-UP

To begin *TIC TAC GO*, you must first load Advanced BASIC into your PC by typing:

```
BASICA ↵
```

Then put your work copy of your *PC Disk* diskette in your default drive and type:

```
RUN "TTG ↵
```

GAME OPTIONS

TIC TAC GO provides several options that affect the difficulty of the game. Your first choice is the size of the pattern you want to use—Small, Medium, Large, Jumbo, or Colossal. These sizes are numbered in order from 1 to 5. To make your selection, simply press the corresponding number on the keypad or keyboard. In general, the smaller the design, the easier it is to match the patterns.

With the second option, choose whether you want to play with the same type of pattern throughout the game or with different designs from round to round. The game is generally easier if you see the same pattern type over and over.

The third option lets you select your player ability level. The beginner level displays simple patterns, while the advanced level has complex, densely-packed patterns.

Similarly, you can control the complexity of the game. Choose an easy, medium, or hard level of difficulty by pressing the number that corresponds to your desired level.

The next feature is referred to as the "no-wipe" condition. You may either have the program remove ("wipe") one pattern from the grid after each deal, or choose to retain all nine patterns for the entire round. It is more difficult to play with the no-wipe condition because it means that you must always try to match the target pattern from among nine grid patterns. As you will soon learn, this new feature will keep you alert throughout the round. To select the no-wipe condition, and play with only nine-pattern deals, press "Y" for "yes" when prompted. If you play without this condition, the number of patterns in the grid will decrease by one with each deal.

The next feature is the "no-match" option. When this condition is active, it is possible for a false deal, that is, one in which *none* of the patterns in the grid match the target pattern. If there is no match, you must press:

```
0
```

before time runs out. If you fail to press zero before time runs out, you will lose points from the round score. On the other hand, if you correctly identify a no-match deal, you receive a double bonus. First, you score twice as much as you would for a regular match. Secondly, a no-match deal does not count as one of the eight deals in the round, so it is a chance to earn extra points and increase your chances of reaching the goal score on any round. The no-match deal appears randomly and can occur several times within a round.

Finally, you can decide whether to have sound accompany the game.

Finally, you can decide whether to have sound accompany the game.

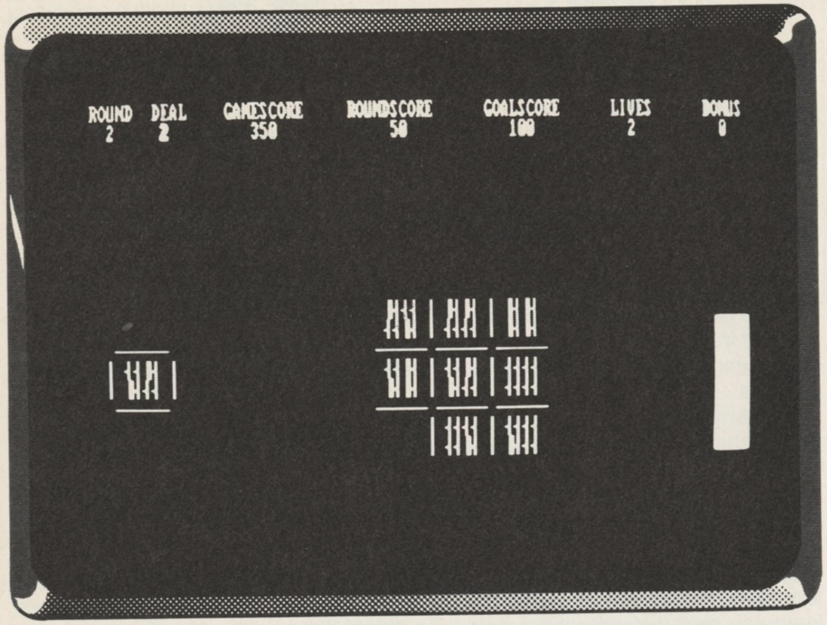
After setting the options, you are asked if you wish to make any changes in your option settings. If you press "N", then the game will begin with those settings. If you press "Y", the options will appear again for alterations. To keep any option at its pre-selected level, press the Enter key. For any option you wish to change, simply press the appropriate key.

At any time, press:

Q ↵

to start the game with whatever settings are currently in place. This option may be used to start a new game even in the middle of a current game.

PLAYING THE GAME



ROUND 2 DEAL 2 GAMESCORE 350 ROUNDSCORE 50 GOALSCORE 100 LIVES 2 BONUS 0

Which pattern is the correct match?

The game is divided into rounds each consisting of eight deals. A deal occurs when the Game Screen displays a single target stimulus pattern and a set of two to nine patterns placed in the *TIC TAC GO* grid.

Each target pattern is presented once, for a specified length of time (indicated by the vertical bar on the screen). You must beat the clock and make the correct match to score points. Use the numeric keypad which, as you will notice, echoes the arrangement of the nine grid locations. For example, to indicate a pattern in the upper right corner, you would press:

9

on the keypad. When playing this game on the IBM PCjr, you would press the same number key from the top of the keyboard.

In the first deal, you will have to match the target from among nine patterns in the grid. If you are correct (or after the time is up), this target and the matched pattern will remain alone on the screen for an instant to indicate the correct match. If you select an incorrect pattern, then both the matched pattern and your wrong selection will remain on the screen briefly so you can see where you made your error.

In the second deal, the new pattern is selected from among the eight patterns left unmatched by the first deal (unless you chose the no-wipe option). In each successive deal, the target pattern is selected from the remaining patterns until the eighth or last deal of the round when the target pattern will have to be matched to only one of two remaining patterns in the grid. If you have reached the goal score on the round, a new round begins once again with eight new deals.

SCORING

At the top of the Game Screen are seven designations to help you keep track of your score and other critical game milestones. From left to right, they are as follows:

Round: the number of the round in the current game
Deal: the number of the deal in the current game
Game Score: your total score, accumulated across all deals and rounds, in the current game
Round Score: your score for the round you are currently playing
Goal Score: the score you must equal or exceed in the current round in order to proceed to the next round. Failure to achieve the goal score terminates the game at the end of the current round.
Lives: the number of chances you have to make the wrong selection. It starts at three and counts down to zero (after three errors). After the fourth and last error, it will read -1, indicating that the game is over. You earn an extra life for every 10,000 points you score.
Bonus: the extra points you can achieve for any extra time remaining on the vertical time bar when you make your selection. The bonus is introduced in later, more difficult rounds.

In general, you score 50 points for a correct match. Since there are eight matches per round, you can score up to 400 points on a round, excluding the bonus. The goal score on the first round is 50 points, and gets progressively higher in later rounds. Also, the allotted time is shortened in later rounds. These are two factors which make the game gradually more challenging.

A SPECIAL BONUS

Beginning with the fourth round of the game, your bonus points come into play. The number of possible bonus points is determined by dividing the total goal score by the amount of time elapsed in the round. This number is then rounded to the nearest multiple of five.

EXITING

At any time, you may exit from *TIC TAC GO* and return to Advanced BASIC by pressing the ESCAPE key.